

Enginyeria en Informàtica
Universitat de Lleida

**Disseny, desenvolupament i integració d'una nova capa
presentació mitjançant Google Web Toolkit per a l'aplicació
d'innovació docent InnovaCampus**

Director: Josep Maria Ribó i Balust
Autors: Ramon Segarra Pijuan i Marcel·lí Alet Alis

Setembre 2010

En record del nostre amic

Francesc Espasa i Palau

Agraïments:

A les mares, pares, Germans i resta de família.
A tots els que han demostrat ser els nostres amics,
per escoltar-nos sempre que ho hem necessitat.
Al Josep Maria Ribó, sense el qual no hauria
sigut possible aquest projecte.

Abstract

InnovaCampus is a teaching application which provides to the educational community an online self-assessment tool for college students. This tool allows students to assess their knowledge and teachers analyze the results to draw the appropriate conclusions.

The tool has evolved over the years, and has recently undertaken a redesign of the application to obtain an application using a modular three-tier architecture that facilitates future extensions.

The persistence and business layers have been developed in the first phase of the redesign and the presentation layer is what gives rise to this project. Thus, the project involves designing, developing and integrating a new presentation layer for the application, in order to provide an interactive, fast and efficient interface to obtain an user-friendly interface using advanced visual properties. Everything is to be achieved by integrating the Google Web Toolkit framework with the created structure using the Spring framework and JPA.

Resum

InnovaCampus és una aplicació d'innovació docent que proporciona a la comunitat educativa una eina online d'autoavaluació per a estudiants universitaris. Aquesta eina permet als estudiants avaluar els seus coneixements i als professors analitzar-ne els resultats per tal d'extreure'n les corresponents conclusions.

L'eina ha evolucionat any a any, i recentment s'ha dut a terme un redisseny de l'aplicació per tal d'aconseguir una aplicació modular utilitzant una arquitectura de tres capes que afavoreixi a futures ampliacions.

Les capes de persistència i de negoci han estat desenvolupades en una primera fase del redisseny i la capa de presentació és la que dona lloc al present projecte. Així doncs, el projecte consisteix en dissenyar, desenvolupar i integrar una nova capa de presentació de l'aplicació, amb la finalitat de proporcionar una interfície interactiva, àgil i eficient per tal d'obtenir un entorn amigable amb propietats visuals avançades. Tot plegat es pretén aconseguir integrant el framework Google Web Toolkit a l'estructura ja creada mitjançant el framework Spring i JPA.

Índex

1	Introducció al projecte InnovaCampus	11
1.1	Antecedents: El projecte InnovaCampus	11
1.1.1	Història	12
1.1.2	Millors en l'enginyeria del software	12
1.2	Objectius	12
1.3	Estructura de la memòria	13
2	Anàlisi de requisits	15
2.1	Anàlisi dels usuaris i l'entorn	15
2.1.1	Estudi de l'audiència	15
2.1.2	Estudi de l'entorn	16
2.1.3	Disseny per la diversitat	17
2.2	Anàlisi del software InnovaCampus	22
2.2.1	Requeriments actuals d'InnovaCampus	22
2.2.2	Aspectes millorables i noves funcionalitats	24
2.3	Requeriments generals i actors del sistema	24
2.4	Casos d'ús	30
2.5	Model de domini	30
2.6	Disseny de la Base de Dades	33
3	Arquitectura de l'aplicació	36
3.1	Introducció a l'arquitectura web	36
3.1.1	Arquitectura Client - Servidor	36
3.1.2	Aplicacions Web	37
3.1.3	HTTP: HyperText Transfer Protocol	37
3.1.4	Aplicacions RIA: El nou paradigma en el desenvolupament d'aplicacions web.	38
3.2	El patró Model Vista Controlador	39
3.2.1	Introducció als Patrons de Disseny	39
3.2.2	El patró Model Vista Controlador (MVC)	40
3.2.3	Introducció al framework SPRING MVC	41
3.3	Arquitectura InnovaCampus	42
3.3.1	La capa de persistència	42
3.3.2	La capa de negoci	43
3.3.3	La capa de presentació	44
3.3.3.1	Aplicacions d'Esriptori vs. Aplicacions Web	44
3.3.3.2	La capa de presentació d'InnovaCampus	44
3.3.3.3	Aportacions d'AJAX a la capa presentació	45

4	Disseny de la capa de presentació	47
4.1	Visió general de Google Web Toolkit (GWT)	47
4.1.1	Perquè GWT ?	47
4.1.2	Arquitectura GWT	49
4.2	Desenvolupament d'aplicacions RIA mitjançant GWT	50
4.2.1	Desenvolupament de la interfície d'usuari	50
4.2.2	Comunicació dels widgets amb el servidor	52
4.2.2.1	RPC Invocació a procediments remots	52
4.2.2.2	Creació d'un servei RPC mitjançant GWT	53
4.2.2.3	Crides a procediments remots RPC	55
4.3	La serialització	56
4.3.1	Tipus de variables serialitzables	56
4.3.2	Classes serialitzables definides per l'usuari	56
4.3.3	Polimorfisme	56
4.3.4	Tipus d'arguments	57
4.4	Integració de GWT amb Spring Framework	59
4.4.1	Com funciona Spring Framework ?	60
4.4.2	Creació d'un exemple simple amb Spring Framework	61
4.4.3	Integració dels serveis GWT amb Spring	62
4.5	Particularitats de GWT en el projecte	68
4.5.1	Gestió de pantalles mitjançant GWT	68
4.5.2	Aplicació del patró singleton	69
4.5.3	Connexió amb la capa de negoci	69
4.5.4	Gestió de l'idioma mitjançant GWT + i18n	71
4.5.5	Extensions de GWT utilitzades	74
4.5.6	El paper del Tiles	74
5	Disseny: Altres aspectes rellevants	77
5.1	Seguretat	77
5.1.1	Seguretat a InnovaCampus: SpringSecurity	78
5.1.1.1	Control d'autoritacions	78
5.1.1.2	Procés d'autenticació	79
5.1.2	Altres aspectes	79
5.1.2.1	La codificació de la clau d'accés	79
5.1.2.2	Control del registre d'estudiant	79
6	Disseny de la interfície gràfica	81
6.1	Anàlisi jeràrquic de tasques (HTA)	81
6.2	Arquitectura de la informació	84
6.2.1	Identificació d'objectius	84
6.2.2	Ordenació de Targetes (Card-Sorting)	85
6.2.3	Estructura i organització de continguts	89
6.2.4	Navegació i Pantalles	90
7	Desenvolupament en comunitat	94
7.1	Desenvolupament en comunitat i llicències	94
7.2	El desenvolupament d'InnovaCampus	97
7.2.1	Eines emprades	97
7.3	Planificació del projecte	98

8	Conclusions i treball futur	101
A	Altres tecnologies emprades	103
A.1	El llenguatge HTML	103
A.2	El llenguatge JavaScript	103
A.3	El llenguatge CSS	104
A.4	El llenguatge XML	105
A.5	El llenguatge JAVA	106
A.6	AJAX	106
A.7	El llenguatge JSP	107
A.7.1	El servlet	108
A.7.2	Java Server Pages	109
A.8	Spring Framework	109
A.9	JPA	110
A.10	SVN	111
B	La capa de persistència	112
B.1	ORM: JPA	112
B.2	Patró DAO	115
B.3	Migració de tecnologies	116
B.4	Transaccions	117
C	La capa negoci	119
C.1	Classes del domini	119
C.2	Els subsistemes	120
D	Casos d'ús	122
D.1	Casos d'ús comuns a tots els usuaris	122
D.2	Casos d'ús específics de professors i alumnes	124
D.3	Casos d'ús específics de l'administrador	127
D.4	Casos d'ús específics dels professors	130
D.5	Casos d'ús específics dels alumnes	148
E	Manual de desplegament	150
E.1	Requisits mínims	150
E.2	Instal·lació del servidor Apache Tomcat	151
E.3	Instal·lació de PostgreSQL	151
E.4	Instal·lar JAVA	152
E.5	Instal·lar SDK de Google Web Toolkit	152
E.6	Instal·lar client subversion	152
E.7	Descarregar i Compilar el codi font de l'aplicació	153
E.8	Importació de la BD's de l'aplicació	153
E.9	Configuració de l'aplicació	153
F	La llicència GPL	155
F.1	Preàmbul	155
F.2	Termes i condicions per la copia, modificació i distribució	156
F.3	Absència de garanties	158
F.4	Final dels termes i condicions	159

Índex de figures

2.1	Estadístiques de navegadors mes a mes (w3schools)	18
2.2	Estadístiques de resolucions mes a mes (w3schools)	19
2.3	Estadístiques de resolucions superiors (w3schools)	20
2.4	Diagrama de casos d'ús de l'administrador	26
2.5	Diagrama de casos d'ús del professor	27
2.6	Diagrama de casos d'ús del professor	28
2.7	Diagrama de casos d'ús de l'alumne	29
2.8	Model de domini	31
2.9	Model Entitat / Relació de la Base de Dades	34
3.1	Arquitectura Client - Servidor	37
3.2	Arquitectura patró MVC	40
3.3	Arquitectura InnovaCampus de tres capes	42
3.4	Model clàssic d'Aplicació Web (Comunicació sincron)	46
3.5	Model AJAX d'Aplicació Web (Comunicació asincron)	46
4.1	Eines de GWT	49
4.2	Arquitectura Client / Servidor amb GWT	50
4.3	Codi GWT del widget de modificació de contrasenya	51
4.4	Aspecte del widget de modificació de contrasenya	52
4.5	RPC - Components de la invocació de procediments remots	53
4.6	RemoteService	53
4.7	RemoteServiceServlet	54
4.8	RemoteService Asincron	54
4.9	Crida al procediment remot de canvi contrasenya	55
4.10	Arguments per a la serialització	57
4.11	Arguments per la serialització	58
4.12	Arquitectura d'InnovaCampus amb GWT	59
4.13	MVC mitjançant Spring Framework	60
4.14	Mapeig del web.xml	61
4.15	Mapeig del Dispatcher-Servlet	61
4.16	El Controlador homeController	61
4.17	Redireccionament peticions rpc	62
4.18	Mapeig peticions controlador GWT	62
4.19	Diagrama integració GWT + Spring	63
4.20	Codi GWTController	64
4.21	Comunicació indirecta	65
4.22	Codi classe GWTController amb comunicació directa	66

4.23	Codi classe SessionAware	66
4.24	Codi classe ServeiImpl	66
4.25	Comunicació directa	67
4.26	Gestió de pantalles i widgets	68
4.27	RemoteServiceImpl.java	70
4.28	RemoteServices.java	70
4.29	RemoteServiceAsync.java	70
4.30	Modul i18N	71
4.31	ICLangConst.java	72
4.32	ICLangMessg.java	72
4.33	Package Lang d'InnovaCampus	72
4.34	Fixters de propietats	73
4.35	Exemple HTM amb etiquetes Tiles	74
4.36	Definició d'un templates	75
4.37	Definició de subtemplates	75
4.38	Definició de subtemplates	76
4.39	Configuració dels Tiles	76
5.1	Interceptors SpringSecurity	78
5.2	Context-param per restringir domini de correu	80
6.1	HTA Afegir estudiant	82
6.2	HTA Afegir tema	82
6.3	HTA Esborrar tema	83
6.4	HTA Modificar tema	83
6.5	Cluster cardsorting	86
6.6	Menús antics i Menús actuals	87
6.7	Menús antics i Menús actuals	88
6.8	Part superior	90
6.9	Part inferior pantalla inicial	90
6.10	Pantalla de l'alumne	91
6.11	Menu d'assignatures	92
6.12	Pantalla del professor	92
6.13	Pantalla de l'administrador	93
7.1	Diagrama de Gantt de les tasques del projecte	99
7.2	Temporalitat total del projecte	100

Capítol 1

Introducció al projecte InnovaCampus

La necessitat de constatar els coneixements adquirits en un procés d'aprenentatge és un procés ineludible tant per a professors com per als propis alumnes interessats en conèixer el seu progrés.

Les noves tecnologies aplicades a l'educació i particularment l'ús d'Internet, permeten la creació de qüestionaris tipus test interactius que són molt útils a l'hora de motivar a l'alumne en tasques d'aprenentatge i de permetre que sigui el propi estudiant, qui constati si ja posseeix els coneixements sobre els quals el professor l'avaluarà posteriorment. De la mateixa manera, aquest tipus d'eines poden ser d'ajut per als professors per saber el grau d'assoliment dels continguts per part dels estudiants i els aspectes més complexos dels propis continguts.

La incorporació de les noves tecnologies en l'ensenyament amb l'objectiu de fer l'estudiant més participatiu, adquireix, encara una major importància en el marc del procés de convergència europea de titulacions universitàries (les conegudes directrius de Bolònia).

Aquest nou model d'ensenyament universitari busca una major consciència i un major protagonisme per part de l'estudiant en l'acte d'aprenentatge. Aquest projecte pretén doncs, oferir a la comunitat educativa (tant a professorat com a estudiantat) una eina lliure amb la finalitat d'ajudar a ambdues parts de la comunitat en el marc educatiu actual.

1.1 Antecedents: El projecte InnovaCampus

Actualment Internet és una eina indispensable en el model educatiu actual, tot aprofitant l'atractiu que ofereix la xarxa, el projecte InnovaCampus planteja una eina de campus virtual que permet als estudiants dur a terme tasques d'autoavaluació de les seves assignatures, i al professorat obtenir informació en forma d'estadístiques per tal d'analitzar el grau de coneixement dels alumnes.

La finalitat d'InnovaCampus és molt bona, però es tracta d'una aplicació desenvolupada com a tres aplicacions completament independents, fet que en fa molt costoses les millores i el manteniment de l'aplicació web. Aquest projecte sorgeix doncs, de la necessitat de disposar d'una aplicació escalable, modular, ben estructurada, ben documentada i preparada per a ser desenvolupada cooperativament com una aplicació de codi lliure.

1.1.1 Història

InnovaCampus sorgeix de la iniciativa del professor Joan Ribera i Calvet, degà de la Facultat de Medicina de la Universitat de Lleida, que necessitava una aplicació amb els objectius de, per una banda oferir una eina d'autoavaluació pels estudiants, i per l'altra permetre al professorat el seguiment dels resultats dels seus alumnes.

Primerament es desenvolupà un prototipus bàsic inicial, aviat se'n va implementar un prototipus millorat afegint-hi funcionalitats i amb una primera implementació del patró controlador (vegeu [1]), més endavant es va millorar l'aplicació amb el disseny de dos nous mòduls, un primer mòdul que tractava la dependència entre preguntes i permetia afegir recursos (imatges, vídeos, ...) a les mateixes (vegeu [2]), i un segon mòdul que incorporava un nou motor de consulta de resultats i generació d'estadístiques (vegeu [3]). Aquests darrers mòduls també incorporaven una enginyeria del software basada en Struts, el patró DAO i el patró de disseny MVC.

1.1.2 Millores en l'enginyeria del software

D'aquest projecte passen per refer totalment InnovaCampus per tal que deixin de ser tres aplicacions independents i passin a ser una única aplicació implementada amb una arquitectura de tres capes, sobre el frameworks Spring, JPA i Google Web Toolkit.

Caldrà que sigui una aplicació molt ben estructurada i molt ben documentada per a futures ampliacions, particularment pensada per al desenvolupament en comunitat.

Primerament és necessària una reenginyeria, des de la reestructuració, que passa per redissenyar la base de dades, replantejant el model de domini, i reimplementar tant les classes de serveis com la interfície. Seguidament cal replantejar la capa de presentació per tal d'adaptar-la a les possibilitats que ens ofereix la tecnologia Web avui en dia.

El primer pas ja ha estat desenvolupat (vegeu [4]), el segon pas, que consisteix en el disseny, desenvolupament i integració d'una nova capa presentació és el que dóna lloc al present projecte.

1.2 Objectius

L'objectiu és continuar la tasca començada pels nostres companys en el redisseny de la capa de persistència per tal de dissenyar, desenvolupar i integrar la nova capa de presentació. Tot plegat ha de permetre continuar el desenvolupament en comunitat i de forma modular com un projecte de codi obert.

És pretén aconseguir una aplicació completament funcional que englobarà totes les funcionalitats que es descriuran en capítols posteriors de la memòria. En particular, aquest projecte pretén:

1. Millorar alguns aspectes d'enginyeria del software:
 - (a) Integració dels frameworks Spring, JPA i Google Web Toolkit (GWT)
 - (b) Correcció d'errors i desenvolupament de millores de la capa de persistència.
 - (c) Proporcionar un desacoblament complet entre capes
2. Proporcionar una aplicació més modular que permeti ampliacions futures i en comunitat senzilles.
3. Proporcionar una aplicació prometedora que pugui ser millorada constantment mitjançant el desenvolupament de l'aplicació en comunitat.
4. Desenvolupar un test complet de l'aplicació abans de ser posat en funcionament.

Per tot plegat es precisa:

- Un disseny elegant de la solució implementada.
- Documentació complerta per desplegar l'aplicació i Javadoc per totes les classes implementades.
- Control de versions enfocat al desenvolupament en comunitat.

1.3 Estructura de la memòria

La memòria d'aquest projecte està estructurada en capítols i seccions. A continuació exposarem un breu resum de què conté cada capítol:

- **Capítol número 1:** Conté una breu descripció del projecte realitzat, d'on partim, quins són els objectius que hem establert a partir de les millores que s'han de desenvolupar.
- **Capítol número 2:** Ens centrem en l'anàlisi de requisits, tan pel que fa a l'anàlisi dels usuaris de l'aplicació i l'entorn com l'anàlisi del software InnoVaCampus.
- **Capítol número 3:** S'explica l'arquitectura de l'aplicació web InnoVaCampus tant en la seva versió anterior com en la que obtindrem una vegada finalitzat el aquest projecte.
- **Capítol número 4:** S'exposa de forma detallada la feina que s'ha realitzat en els processos de disseny, integració i desenvolupament de la nova capa presentació.
- **Capítol número 5:** Es presenta altres aspectes rellevants del desenvolupament com ara la seguretat, el control d'autoritzacions, ...
- **Capítol número 6:** S'explica el procés de disseny de la interfície gràfica pel que fa a estructuració de continguts i aspecte de la interfície.
- **Capítol número 7:** Expliquem quin ha estat el procés de desenvolupament del projecte InnoVaCampus fins l'actual versió.
- **Capítol número 8:** Conclusions del projecte i futurs treballs i millores d'InnoVaCampus.
- **Annex A:** S'expliquen les tecnologies utilitzades en el desenvolupament d'aquest projecte.
- **Annex B:** Conté un resum de la capa de persistència.
- **Annex C:** Conté un resum de la capa de negoci.

- **Annex D:** Conté el detall de cada un dels casos d'ús d'InnovaCampus.
- **Annex E:** Conté les especificacions i explicacions del desplegament de l'aplicació web InnovaCampus.
- **Annex F:** Conté la llicència GNU-GPL que utilitza InnovaCampus.

Capítol 2

Anàlisi de requisits

L'anàlisi de requisits és el primer pas en el desenvolupament d'una aplicació, ja que abans d'iniciar qualsevol part de la nostra aplicació necessitem establir els serveis que el sistema ha de proporcionar i les restriccions sota les quals ha d'operar.

En el nostre cas es realitza mitjançant una anàlisi del context de l'aplicació actual i del model definit en el desenvolupament de la capa de persistència, tot identificant un model de casos d'ús, que consisteix en identificar quines funcionalitats i restriccions ha de proporcionar l'aplicació en cada cas, extraient el corresponent model de domini.

2.1 Anàlisis dels usuaris i l'entorn

En aquesta fase ens centrarem bàsicament en l'estudi dels usuaris potencials de l'aplicació i l'entorn en el que l'utilitzaran, ja que així podrem ser molt més precisos en el disseny posterior.

Al nostre entendre és important saber bé sobre què estem treballant, de manera que conèixer més i millor el context en el que ens movem, ens ajudarà a obtenir un producte millor, ampliant l'accessibilitat al major nombre d'usuaris i aconseguint una alta usabilitat del producte.

2.1.1 Estudi de l'audiència

En aquest punt és important definir el segment de mercat al qual volem accedir (edat, sexe, educació, ocupació, ...) i el nivell d'experiència que necessitaran aquests usuaris per poder gaudir i sobretot que els resulti útil l'aplicació que s'ha de desenvolupar.

L'aplicació InnovaCampus es centra principalment en un sector de població comprès entre els 18 i 65 anys, tenint en compte que l'aplicació és accessible per qualsevol membre de la Universitat ja siguin PDI o Alumnes. Podem doncs afirmar que qualsevol persona s'hi pot encabir, sempre que tingui uns mínims coneixement previs tant informàtics com d'Internet.

Anem doncs a estudiar les diferents franges d'edat del usuaris d'InnovaCampus:

- Dels 18 anys fins als 28 anys.
 - Usuaris amb rol d'estudiant.
 - Aquests usuaris utilitzaran l'aplicació per a realitzar els test de les diferents assignatures a les quals estiguin matriculats.
 - La major part d'ells disposen de nivells d'experiència suficients per a l'ús sense dificultats de l'aplicació i al mateix temps disposen de les infraestructures necessàries per fer-ne us. Hem de tenir en compte que molts d'ells ja han tingut experiències prèvies amb aplicacions e-Learning a l'institut i són grans coneixedors i usuaris d'eines d'internet.
- Aparti de 30 anys.
 - Usuaris amb rol de professor.
 - Aquests usuaris utilitzaran l'aplicació per a avalua el progrés dels alumnes en les assignatures que imparteixen mitjançant proves escrites en format test.
 - Hem de tenir en compte que aquest és un sistema d'avaluació que els professors que ho requereixin poden utilitzar, per tant no és obligatori i creiem que els que l'utilitzin ho faran perquè ja han experimentat experiències prèvies amb aplicacions paregudes.

Un cop hem identificat els dos segments de mercat de la nostra aplicació, relacionant l'edat dels usuaris amb el rol que tindran a l'aplicació, és el moment de determinar quines són les possibles dificultats amb les que es poden trobar aquests usuaris alhora d'utilitzar InnovaCampus.

Volem que la nostra aplicació sigui acceptada i utilitzada per el major nombre de persones i en el menor temps, per tant creiem que l'aplicació ha de ser el més intuïtiva i fàcil d'utilitzar possible, és a dir, usable i accessible, per tot això, es procurarà no exigir un nivell d'experiència ni de tipus tècnic ni de tipus cognitiu molt elevat, sinó tot el contrari.

2.1.2 Estudi de l'entorn

L'objectiu de l'estudi de l'entorn és el d'aportar la informació referent al lloc o llocs físics on es produirà la interacció entre l'usuari i l'aplicació.

Com bé sabem, InnovaCampus és una aplicació d'innovació docent tant per alumnes com per a professors, per això és important determinar en quins llocs s'utilitzarà l'aplicació, per a que s'utilitzarà i per quin tipus d'usuaris.

- Realització de test (autoavaluació i aprenentatge)
 - Alumnes
 - Des de qualsevol lloc

- Realització de test (puntuables)
 - Alumnes
 - Únicament des de l'aula especificada
- Gestió de dades personals
 - Alumnes i professors
 - Des de qualsevol lloc
- Gestió de les assignatures (temes, preguntes, respostes, ...)
 - Professors i administrador
 - Des de qualsevol lloc
- Visualització d'estadístiques
 - Professors
 - Des de qualsevol lloc

Tal com es pot observar, l'aplicatiu ha de permetre l'accés des de qualsevol lloc, ja que així ho requereixen les principals funcionalitats encara que els tests puntuables han d'estar restringits únicament a les aules específiques.

D'aquí podem extreure els dos possibles entorns de treball, per una banda totes les aules de la Universitat i per l'altra qualsevol PC amb accés a internet.

Per tant haurem de tenir en compte aquestes dades i dissenyar l'aplicació de la forma més adient per tal d'encabir els dos entorns en els que ens podem trobar i no tenir que dissenyar una aplicació per a cada entorn.

2.1.3 Disseny per la diversitat

Els llocs web estan exposats a gent d'orígens ben diferents. Qualsevol persona amb un ordinador connectat a Internet pot visitar l'aplicació, independentment del seu origen geogràfic, cultural o generacional. Per tant, s'intentarà construir el lloc web tenint en compte aquests aspectes i no únicament centrar-nos en l'àmbit més proper.

Un dels requisits principals serà l'idioma, per aquest motiu és requisit indispensable la internacionalització de l'aplicació utilitzant sistemes i18n per tal de fer que aquesta sigui fàcilment traduïble a diferents idiomes. En un inici l'aplicació es trobarà en tres idiomes (Angles, Castellà i Català), tot i que és ampliable a altres idiomes amb prèvia traducció.

Per determinar quines són les possibles i principals dificultats alhora d'utilitzar InnovaCampus les dividirem en:

- Dificultats tècniques: S'estudien les dificultats tècniques com ara navegadors, resolucions, velocitats de connexió, ...
- Dificultats cognitives: S'estudien les dificultats d'aprenentatge i raonament de la informació.

Dificultats tècniques:

La diversitat de navegadors sempre han estat un entrebanc important a l'hora de desenvolupar aplicacions web, ja que en molts casos, desenvolupar aplicacions que funcionin o es comportin de forma idèntica amb els diferents navegadors que existeixen actualment és gairebé una missió impossible, provocat principalment per la no estandardització dels motors de renderització que tracten la informació de les pàgines web i per tant moltes aplicacions han de disposar de parts adaptades a cada un dels navegadors.

Per aquest motiu és important tenir en compte quin serà el navegador amb el que la major part d'usuaris accediran a la nostra aplicació, i poder així gestionar de forma millor aquesta problemàtica.

2010	Internet Explorer	Firefox	Chrome	Safari	Opera
August	30.7 %	45.8%	17.0%	3.5%	2.3%
July	30.4 %	46.4%	16.7%	3.4%	2.3%
June	31.0 %	46.6%	15.9%	3.6%	2.1%
May	32.2 %	46.9%	14.5%	3.5%	2.2%
April	33.4 %	46.4%	13.6%	3.7%	2.2%
March	34.9 %	46.2%	12.3%	3.7%	2.2%
February	35.3 %	46.5%	11.6%	3.8%	2.1%
January	36.2 %	46.3%	10.8%	3.7%	2.2%

Figura 2.1: Estadístiques de navegadors mes a mes (w3schools)

Tal i com s'observa a la figura 3 actualment el navegador més utilitzat és Firefox, seguit d'aprop per Internet Explorer i Google Chrome, per tant seria molt interessant aconseguir un funcionament per igual en aquests tres navegadors.

Pel que fa Safari, hem de tenir en compte que utilitza el mateix motor de renderització que Google Chrome (WebKit).

També és important tenir en compte si són navegadors amb llicència lliure o propietària, ja que això també pot ser un factor decisiu alhora de que els usuaris l'utilitzin.

- **Mozilla Firefox:** Llicència de software lliure
- **Internet Explorer:** Llicència de software propietari
- **Google Chrome:** Llicència de software lliure
- **Safari:** Llicència de software propietari
- **Opera:** Llicència de software propietari

Un altre aspecte a analitzar serà la compatibilitat del navegador amb els diferents sistemes operatius existents actualment, ja que no tots els usuaris accediran a l'aplicació amb el mateix sistema operatiu.

- **Mozilla Firefox:** Windows, MacOS, Linux, BSD, Solaris, GNU
- **Internet Explorer:** Windows
- **Google Chrome:** Windows, MacOS, Linux
- **Safari:** Windows, MacOS, iOS
- **Opera:** Windows, MacOS, Linux, BSD, Solaris, BeOS, Android

Una altre aspecte a tenir en compte és la resolució i els colors de definició de la pantalla amb la que visualitzaran els usuaris l'aplicació.

Date	Higher	1024x768	800x600	640x480	Unknown
January 2010	76%	20%	1%	0%	3%
January 2009	57%	36%	4%	0%	3%
January 2008	38%	48%	8%	0%	6%
January 2007	26%	54%	14%	0%	6%
January 2006	17%	57%	20%	0%	6%
January 2005	12%	53%	30%	0%	5%
January 2004	10%	47%	37%	1%	5%
January 2003	6%	40%	47%	2%	5%
January 2002	6%	34%	52%	3%	5%
January 2001	5%	29%	55%	6%	5%
January 2000	4%	25%	56%	11%	4%

Figura 2.2: Estadístiques de resolucions mes a mes (w3schools)

Fonamentalment la població utilitza resolucions de 1024 x 768 però des de fa gairebé 2 anys i amb l'abaratiment del cost de la pantalles planes LCD, s'ha disparat l'ús de pantalles de fins a 24 polsades

o més les quals utilitzen resolucions molt majors. Per tant no caldrà que ens restringim al desenvolupament 1024 x 768 sinó que podem excedir-nos en la resolució permeten així una major amplitud i al mateix temps una major visualització de totes les parts de la interfície, unificant així diferents seccions en una sola interfície i reduint el nombre de scrolls que hi puguin haver.

January 2010	
Resolution	% of Total
1280x1024	18.2 %
1280x800	17.3 %
1440x900	10.5 %
1680x1050	10.0 %
1920x1200	4.6 %
1366x768	3.6 %
1920x1080	2.3 %
1152x864	2.1 %
1600x1200	1.4 %
1280x768	1.2 %
Other	4.8 %

Figura 2.3: Estadístiques de resolucions superiors (w3schools)

Pel que fa a l'ample de banda de connexió a la xarxa, avui en dia ja no existeix excusa per dir que no tenim una mínima connexió a Internet, ja que les ofertes de les companyies actuals ofereixen a preus acceptables connexions suficientment ràpides com per poder suportar aplicacions amb els últims avenços tecnològics.

Per tant creiem que aquest és un aspecte que només s'ha de tenir en compte per a situacions puntuals en nuclis de població molt reduïts i aïllats.

Actualment l'estat espanyol es troba a la cua dels principals països pel que fa a la qualitat de la banda ampla d'accés a Internet, amb una mitja de connexió 1,2 Mbps a anys llum de països com Estats Units (5,4 Mbps), França (17,6 Mbps), Suècia (18,2 Mbps) o Japó amb una connexió mitjana de 61 Mbps.

Dificultats cognitives:

Els éssers humans som diferents. Totes les interfícies d'usuari haurien d'acomodar-se a aquestes diferències de tal forma que qualsevol persona fos capaç d'utilitzar-les sense problemes. Per tant és necessari evitar dissenyar, només atenent a característiques de grups de població específics, imposant barreres innecessàries que podrien ser evitades posant una mínima atenció a aquestes qüestions.

Si pensem en gent amb dificultats de visió, és important, i ho tindrem en compte, no codificar cap conducta important únicament amb colors, per tant utilitzarem elements gràfics que serveixin de

suport per a complementar el significat com ara l'ús d'icones, imatges o text alternatiu.

Pel que fa als usuaris amb problemes auditius, intentarem no fer del so un mitjà exclusiu i tindrem en compte no codificar certs missatges d'alerta amb sons, ja que els usuaris amb discapacitats auditives poden no percebre el risc associat a certa situació.

Per últim no podem oblidar les persones amb problemes cognitius, en aquest cas hem d'intentar planificar el sistema de tal manera que es basi en la senzillesa evitant interaccions complicades i amb moltes opcions.

Volem que la nostra aplicació sigui acceptada i utilitzada per major nombre de persones i en el menor temps possible, per tant creiem que l'aplicació ha de ser el més intuïtiva i fàcil d'utilitzar possible, és a dir, usable i accessible, procurant no exigir un nivell d'experiència ni de tipus tècnic ni de tipus cognitiu molt elevat, sinó tot el contrari.

2.2 Anàlisi del software InnovaCampus

2.2.1 Requeriments actuals d’InnovaCampus

Com ja hem introduït en el capítol anterior, InnovaCampus és un software innovador que pretén proporcionar una eina a alumnes i professors per tal de generar testos d’autoavaluació i les estadístiques conseqüents.

Passem a descriure les funcionalitat actuals d’InnovaCampus que són les que marcaran els requisits del redisseny de la nova aplicació:

- Canviar password d’usuari
- Recordar clau
- Modificar dades d’usuari
- Registrar-se
- Veure anuncis de l’administrador
- Veure anuncis del taulell d’anuncis d’una assignatura
- Donar d’alta un professor
- Donar de baixa un professor
- Donar d’alta un anunci
- Donar de baixa un anunci
- Donar d’alta una assignatura
- Donar de baixa una assignatura
- Modificar dades d’una assignatura
- Reiniciar una assignatura
- Donar d’alta un professor a una assignatura
- Donar de baixa un professor a una assignatura
- Donar d’alta un alumne a una assignatura
- Donar d’alta un llistat d’alumnes a una assignatura mitjançant un fitxer
- Donar de baixa un alumne d’una assignatura
- Donar d’alta un tema a una assignatura
- Donar de baixa un tema a una assignatura
- Modificar un tema d’una assignatura
- Donar d’alta un objectiu a un tema d’una assignatura

- Donar de baixa un objectiu d'un tema d'una assignatura
- Modificar un objectiu d'un tema d'una assignatura
- Donar d'alta una pregunta a un objectiu
- Donar de baixa una pregunta d'un objectiu
- Modificar una pregunta d'un objectiu
- Donar d'alta una resposta a una pregunta
- Donar de baixa una resposta d'una pregunta
- Donar d'alta un anunci al taulell d'anuncis d'una assignatura
- Donar de baixa un anunci
- Resoldre un test

Funcionalitat de les estadístiques:

- Calcular estadístiques per temes, objectius, preguntes i tests
- Calcular estadístiques per a un alumne concret en temes, objectius, preguntes i tests
- Estadístiques entre dues dates
- Generació dinàmica de gràfiques
- Exportar dades a fulls de càlcul i altres formats
- Estadístiques calculades (percentatges, desviacions típiques, etc.)
- Cercador d'alumnes

Requeriments de l'entorn:

- El sistema ha de poder operar sota qualsevol sistema operatiu
- L'aplicació ha de detectar l'entrada errònia de dades

Requeriments de la interfície:

- El sistema es comunicarà amb altres sistemes mitjançant el protocol HTTP
- Les dades estaran en HTML sempre que sigui possible

Restriccions de disseny:

- Compliment dels estàndards XHTML, CSS i WAI-A
- Ús d'AJAX per a simular una aplicació d'escriptori
- Framework Google Web Toolkit (vegeu [Capítol 4])
- Ús de l'especificació JSP 2.0

2.2.2 Aspectes millorables i noves funcionalitats

A nivell de disseny i implementació, InnovaCampus presenta una sèrie de característiques que haurien de ser millorades.

Hem enumerat les més importants:

1. El disseny dels Beans no era el més elegant. S'ha revisat la seva implementació i disseny per tal d'aconseguir més coherència.
2. La interacció amb el sistema no és gaire còmoda. Així, per exemple, sempre que volem donar d'alta, de baixa o simplement modificar un objectiu, pregunta o resposta hem de seleccionar primer l'assignatura d'una llista. Això és una tasca força repetitiva i gens còmoda. Seria molt més còmode tenir un espai de treball per a cadascuna d'elles i no haver-les de seleccionar per a cada acció que es dugui a terme.

2.3 Requeriments generals i actors del sistema

L'objectiu bàsic del projecte és el de desenvolupar un entorn docent encarat a alumnes i professors per a oferir possibilitats d'autoavaluació als primers i de estadística de progressos i assimilació de conceptes als segons. Tot plegat ho hem d'aconseguir generant una aplicació elegant i que segueixi els patrons de disseny adients.

Fins ara en aquest capítol hem anat fent referència a molts dels aspectes que haurem de tenir en compte. Per no fer-nos repetitius i per fer-ho més entenedor, en aquest punt presentarem els objectius parlant d'aquelles funcionalitats que l'aplicació ha de proporcionar als usuaris de la mateixa.

L'aplicació tindrà tres tipus d'usuaris: L'administrador, els professors i els estudiants.

Presentem doncs, de forma global els objectius més rellevants de l'aplicació, partint de les funcionalitats que ha de proporcionar aquesta i classificant-les segons el rol d'usuari que exerceixen:

Quan es faci referència a la capacitat de gestionar es referirà a la capacitat de donar d'alta, baixa o modificació.

Administrador:

És l'encarregat de control, manteniment i supervisió de tota l'aplicació. En principi l'administrador ha de ser capaç de gestionar tot el que un professor pugui fer.

Entre les seves funcionalitats distingim:

- Donar d'alta/baixa als professors.
- Canviar les seves dades: Login i Password.
- Donar d'alta/baixa anuncis. Els anuncis podran anar dirigits a alumnes, professors, a ambdós, o podran ser anuncis generals per a usuaris i per aquells que no accedeixin com a usuaris.

Professor:

És l'encarregat de gestionar les assignatures que té assignades, podrà actualitzar-les, modificar-les, eliminar-les i fins i tot delegar-les a altres professors per a que es facin càrrec de la seva assignatura, així com també tenen la responsabilitat de matricular o donar de baixa un estudiant en una o varies assignatures.

Entre les seves funcionalitats distingim:

- Canviar les seves dades: Nom, Cognom, Email, Telèfon, Login i Password.
- Pot gestionar assignatures.
- Pot gestionar professors.
- Pot gestionar temes.
- Pot gestionar objectius.
- Pot gestionar preguntes.
- Pot gestionar respostes.
- Permet reinicialitzar una assignatura.
- Permet matricular alumnes, ja sigui fent l'alta manualment des de teclat o mitjançant un fitxer amb un format determinat.
- Pot visualitzar estadístiques que fan referència als diferents tests realitzats pels estudiants d'una assignatura. Les estadístiques podran ser referents a un estudiant concret, o més globals, referents a un objectiu concret d'un tema de l'assignatura.
- Pot gestionar el taulell d'anuncis de l'assignatura.
- Pot visualitzar anuncis de l'administrador.
- Pot afegir qüestions al sistema des d'un fitxer XML.

Alumnes:

Per la seva part, l'estudiant té la finalitat d'autoavaluar-se mesurant els seus coneixements amb l'ajuda de l'aplicació.

Entre les seves funcionalitats distingim:

- Canviar les seves dades: Nom, Cognom, Email, Login i Password.
- Permet resoldre tests i observar-ne els resultats.
- Visualitzar anuncis de l'administrador.
- Visualitzar el taulell d'anuncis de les assignatures.

Mitjançant un diagrama de casos d'ús podem tenir una síntesi dels usuaris i casos d'ús que exerciran.

El corresponent a la nostra aplicació és el següent:

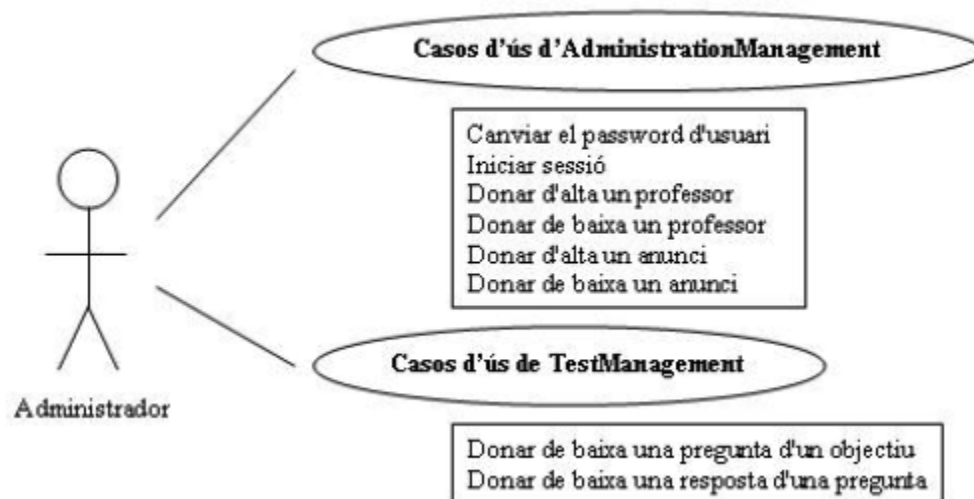


Figura 2.4: Diagrama de casos d'ús de l'administrador

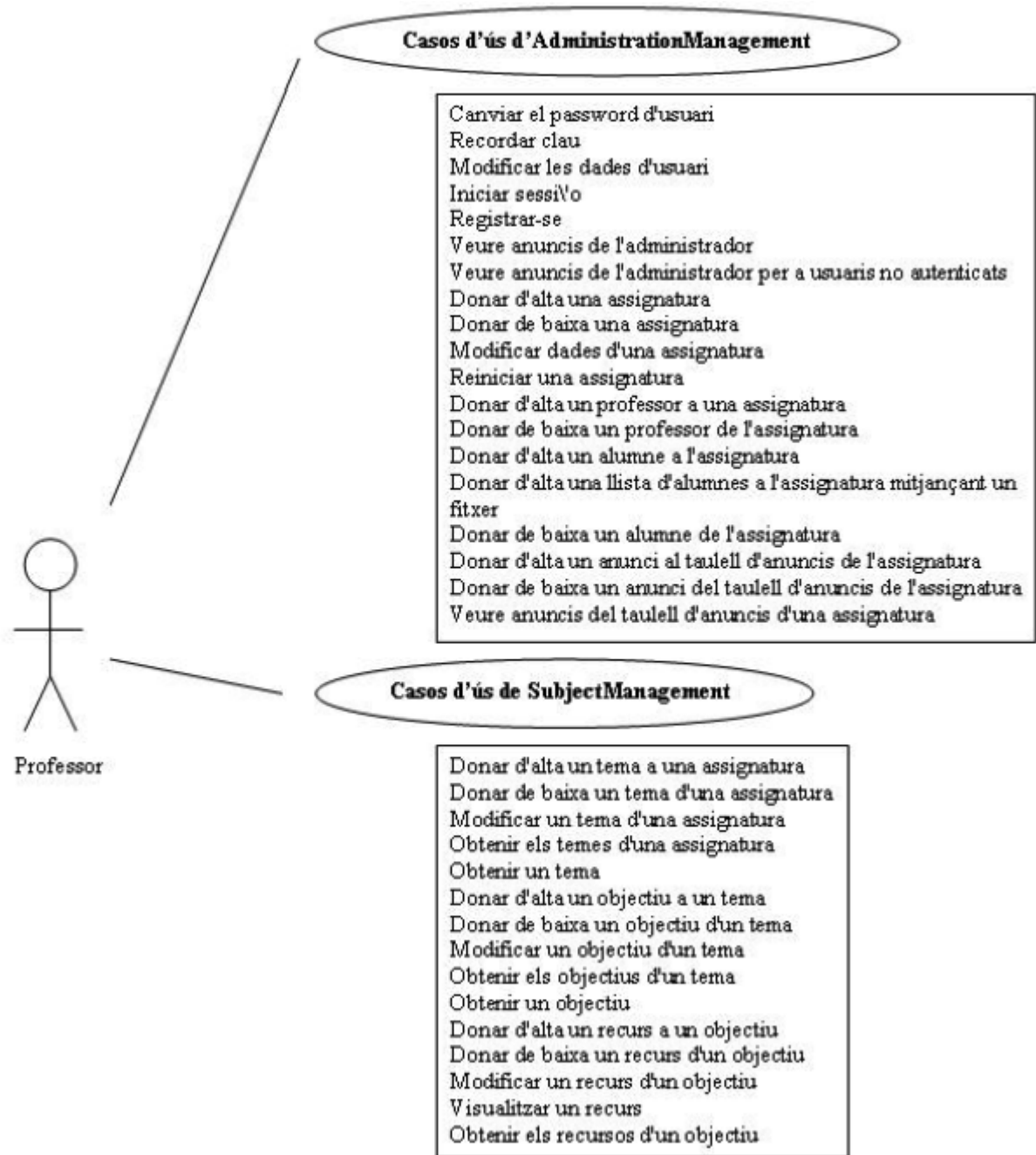


Figura 2.5: Diagrama de casos d'ús del professor

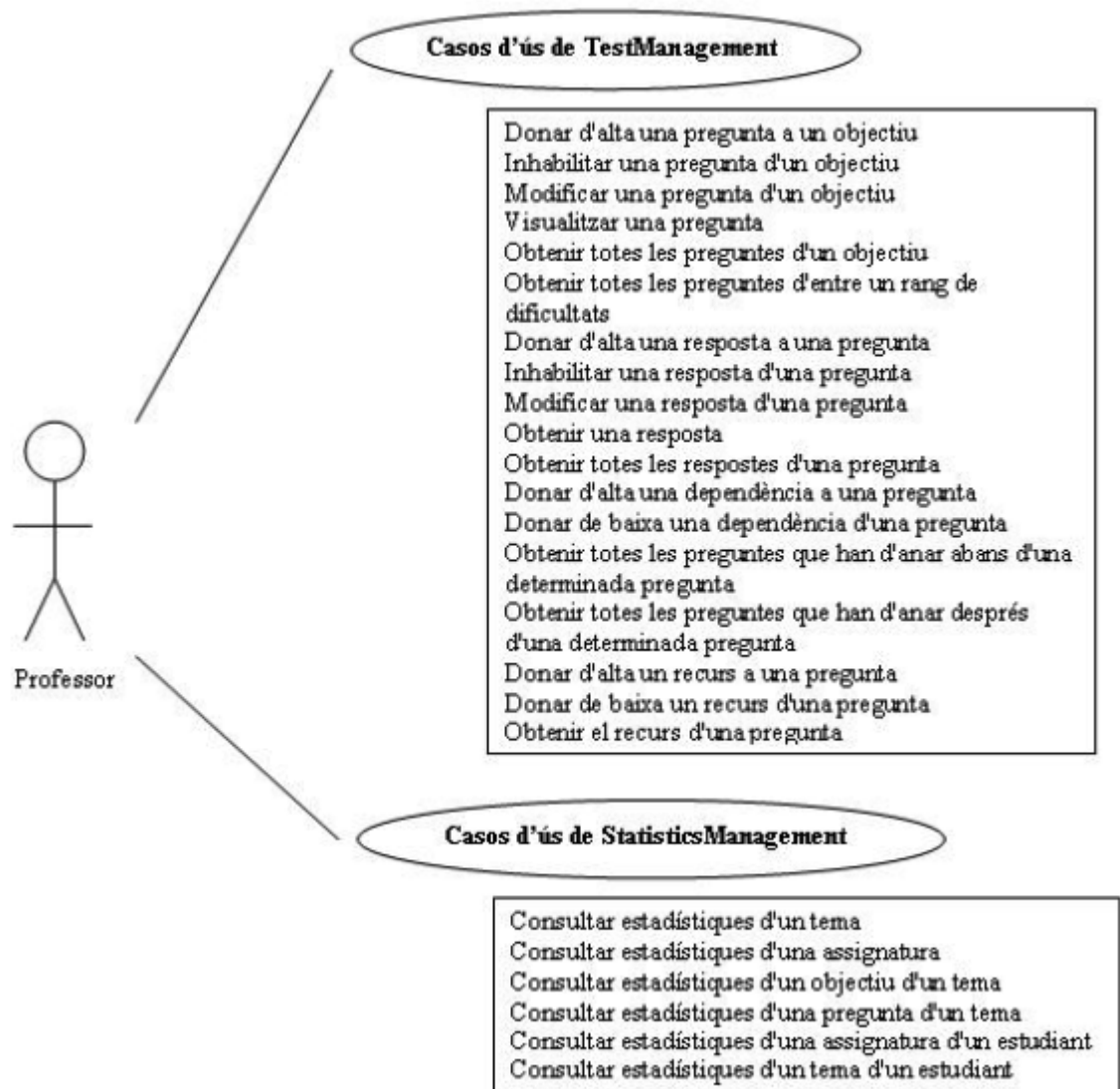


Figura 2.6: Diagrama de casos d'ús del professor

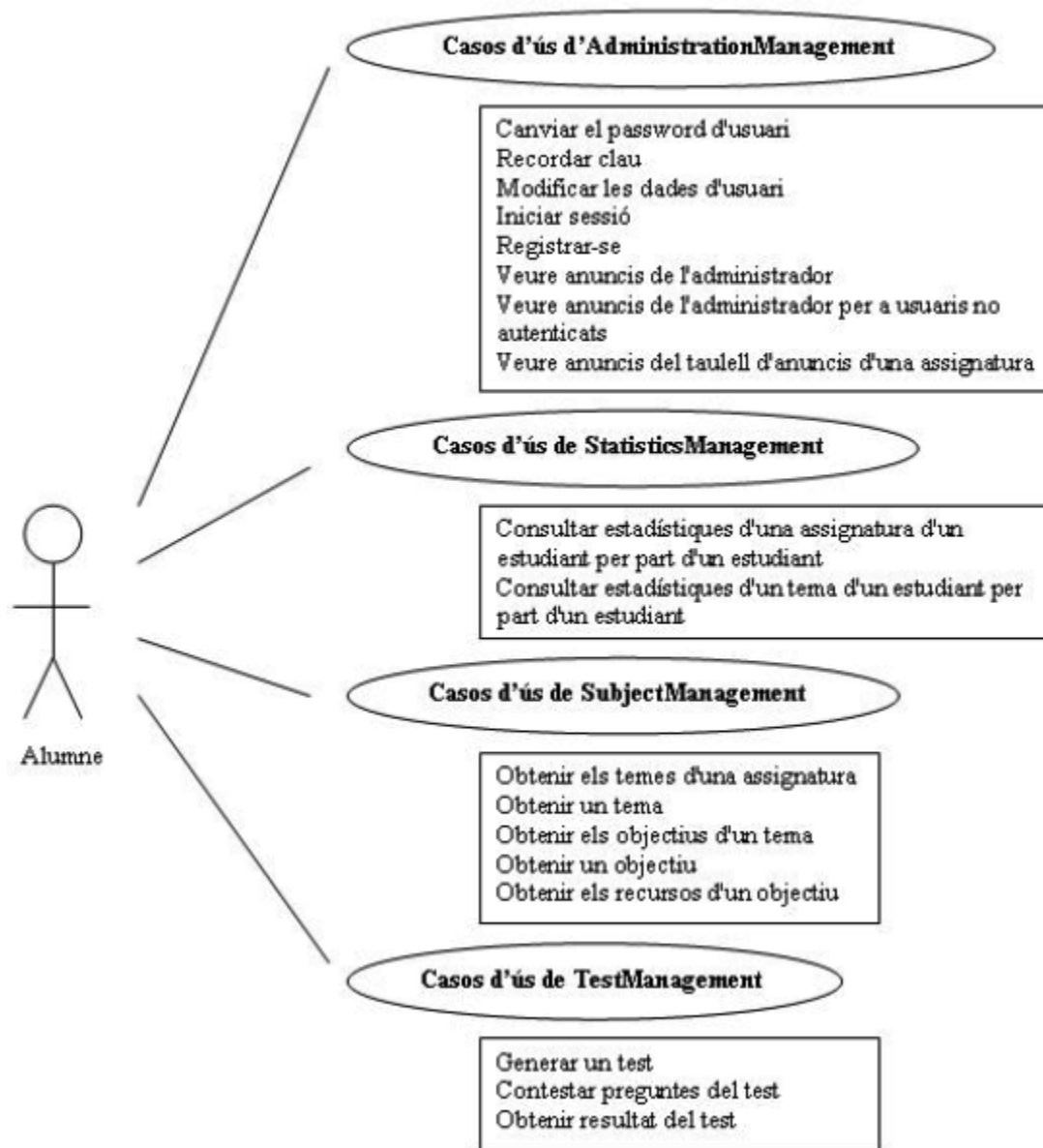


Figura 2.7: Diagrama de casos d'ús de l'alumne

Com podem observar tenim 3 actors bàsics, cadascun d'ells serà un usuari de la nostra aplicació, amb els seus corresponents casos d'ús.

2.4 Casos d'ús

El diagrama de casos d'ús representa la forma com un client (Actor) opera amb el sistema, a més a més de la forma, tipus i ordre en què els elements interactuen. Actor és un rol en què l'usuari interactua amb el sistema, no necessàriament representa a una persona en particular, sinó més aviat la tasca que realitza en front el sistema. El document annex [D] conté l'especificació detallada de tots els casos d'ús classificats segons el rol d'usuari, així com el raonament associat a la necessitat de tenir-los i el seus propòsits. Tot seguit, n'adjuntem un exemple:

Cas d'ús: Canviar password d'usuari

Amb aquest cas d'ús, permetem que els diferents usuaris de l'aplicació, tinguin l'oportunitat de canviar el seu password, oferint així una eina de seguretat. El cas d'ús es desenvolupa de la mateixa manera que es faria en una típica aplicació on es canvia el password, és a dir, primer es demanaria el password antic i després el nou, i un cop validada la informació, el nou password quedaria registrat.

USUARI	SISTEMA
1. L'usuari es connecta a la pàgina i entra el login i password.	2. El sistema registra la informació.
3. L'usuari indica que vol canviar el seu password.	4. El sistema demana el password antic i el nou.
4. L'usuari introdueix el password actual i el nou password.	6. El sistema procedeix a fer la modificació.
7. L'usuari veu que tot ha anat correctament.	

Taula 2.1: Cas d'ús “Canviar contrasenya usuari”

Casos d'error:

1. El login i el password no són correctes. S'ha denegat l'accés a la pàgina i s'informa de l'error.
5. Password no vàlid. No s'efectua la modificació i s'informa de l'error.
5. Password antic incorrecte. No s'efectua el canvi i s'informa de l'error.

2.5 Model de domini

Un cop analitzats els requisits de l'aplicació, cal definir les entitats que hauran de ser modelitzades a l'aplicació així com les interrelacions entre elles.

El model de domini és una representació gràfica, extreta dels casos d'ús, de les entitats que formen el projecte i les relacions que s'estableixen entre elles.

El model de domini obtingut després del redisseny de la capa de persistència és el següent:

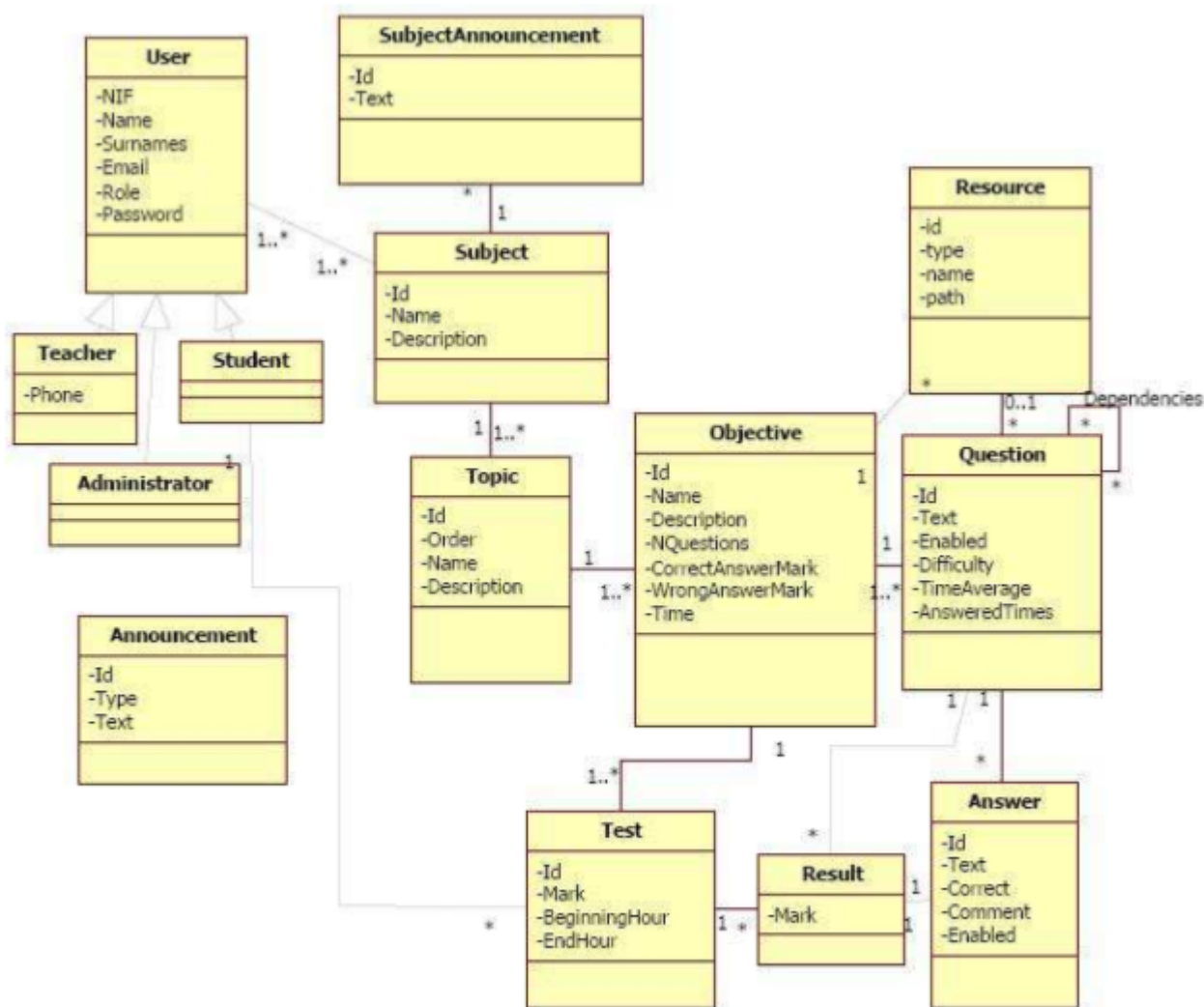


Figura 2.8: Model de domini

Explicació del model dissenyat:

A continuació es descriuran breument les diferents entitats del model de domini:

- **User:** Engloba les entitats Administrator, Teacher i Student, que comparteixen la majoria d'atributs.
- **Administrator:** És l'usuari amb més permisos i l'encarregat de supervisar el sistema.
- **Teacher:** És l'usuari que disposa de més funcionalitats. Cada professor és responsable d'una o varies assignatures a les quals pot afegir preguntes, respostes, consultar-ne les estadístiques, matricular-hi alumnes i publicar-hi anuncis. S'identifica amb el NIF.

- **Student:** És l'usuari que s'encarrega de realitzar testos per autoavaluar-se. Cada estudiant pot estar matriculat a una o vàries assignatures, i pot consultar les seves estadístiques. S'identifica amb el NIF.
- **Announcement:** Aquesta entitat representa els anuncis globals del sistema, ja siguin per a usuaris no connectats, per a professors, per a estudiants o per a tothom, i és la utilitzada per l'administrador.
- **Subject:** Les assignatures estan dividides en temes i poden tenir anuncis associats.
- **SubjectAnnouncement:** Aquest tipus d'anuncis són els que publiquen els professors i únicament i són exclusius de l'assignatura.
- **Topic:** Els temes tenen un ordre i estan dividits en objectius.
- **Objective:** Aquestes entitats emmagatzemen la informació de quantes preguntes formaran un test, quina serà la puntuació aplicada en les respostes correctes i errònies, i el temps màxim per a resoldre un test. A més els objectius poden tenir recursos associats, preguntes i testos.
- **Test:** L'entitat test només gestiona de les dades globals d'aquest: la nota, l'hora d'inici i l'hora de finalització. Cada test té varis resultats associats, un per a cada pregunta que forma el test.
- **Result:** Relaciona una pregunta del test amb la corresponent resposta identificant la nota obtinguda.
- **Question:** Les preguntes tenen un text, poden estar associades a recursos, poden dependre d'altres preguntes i tenen una sèrie de respostes.
- **Answer:** Cada resposta disposa del seu text, un possible comentari establert per part del professor i la indicació de si és una de les respostes correctes.

2.6 Disseny de la Base de Dades

Amb l'objectiu de dotar de persistència les entitats de software s'ha desenvolupat una base de dades relacional obtinguda a partir del model de domini presentat en el punt anterior.

El disseny de la base de dades relacional es mostra a continuació. S'han marcat en negreta els atributs referents a les claus primàries de cada entitat (l'atribut que les identifica) , les claus foranes (les que provenen d'una altra entitat i únicament podran tenir valors que estiguin inclosos en tals entitats) s'han subratllat.

Aquesta és la llista de relacions resultant:

USERS (nif , name, surnames, password, role, email)
TEACHER (nif , phone)
STUDENT (nif)
ADMINISTRATOR (nif)
SUBJECT (id , name, description)
TOPIC (id , name, description, number, <i>subject_id</i>)
OBJECTIVE (id , name, description, nquestions, correctanswervalue, wronganswervalue, maxtime per test, <i>topic_id</i>)
RESOURCE (id , name, path, type, <i>objective_id</i>)
TEST (id , mark, beginningTime, endTime, nif, <i>objective_id</i>)
QUESTION (id , text, difficulty, answeredTimes, timeAverage, <i>objective_id</i> , <i>resource_id</i> , enabled)
ANSWER (id , text, comment, correct, enabled, <i>question_id</i>)
RESULT (<i>test_id</i> , <i>question_id</i> , <i>answer_id</i> , mark)
RESPONSIBLE (<i>subject_id</i> , nif)
ENROLMENT (<i>subject_id</i> , nif)
DEPENDENCIES (<i>question_id_a</i> , <i>question_id_b</i>)
ANNOUNCEMENT (id , text, type)
SUBJECTANNOUNCEMENT (id , text, <i>subject_id</i>)
DIFFICULTY (question_id , nif , <i>subject_id</i> , <i>topic_id</i> , <i>objective_id</i> , correctAnsweredTimes, answeredTimes, student_difficulty)

Taula 2.2: Relacions existents a la Base de Dades

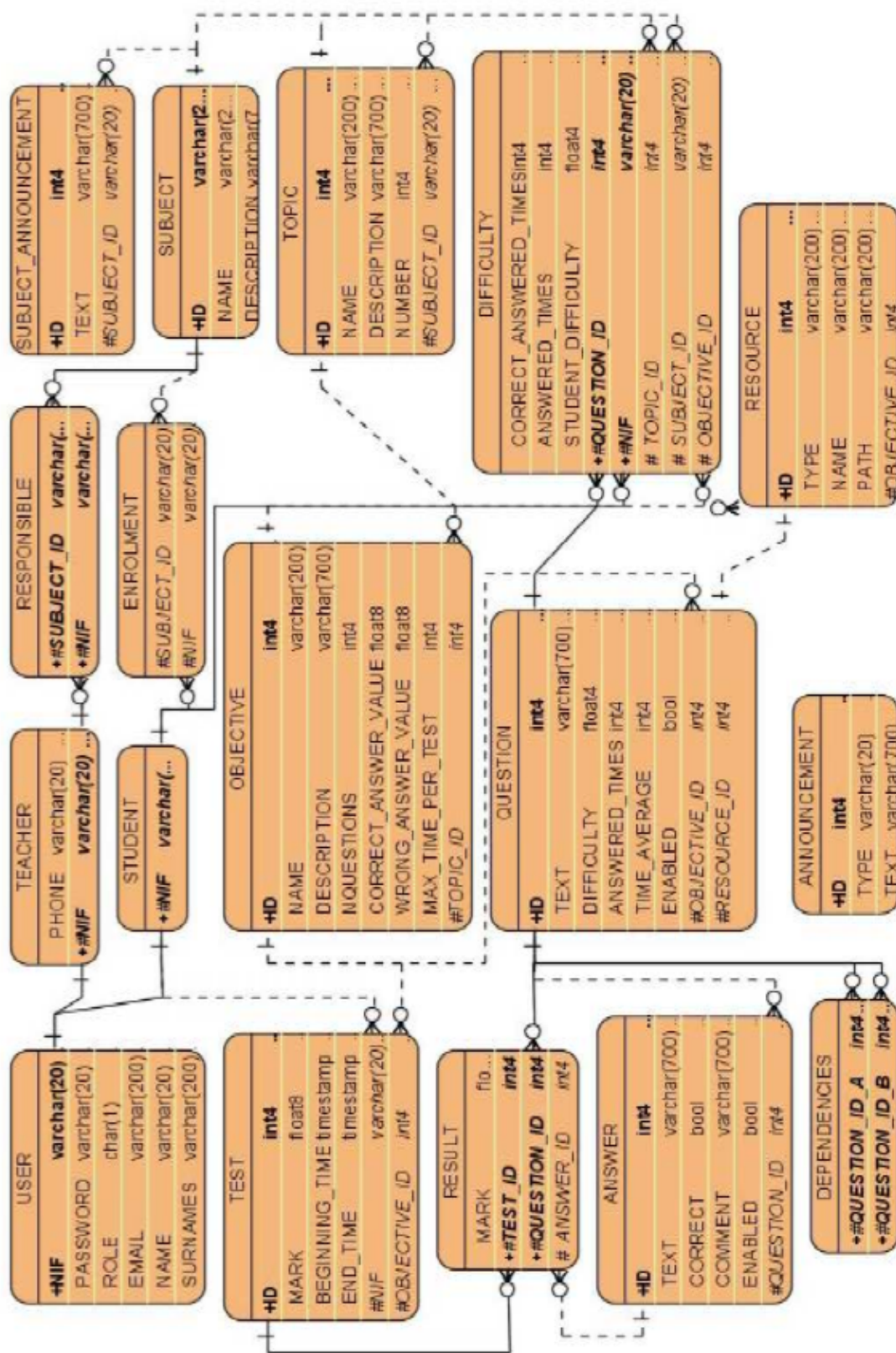


Figura 2.9: Model Entitat / Relació de la Base de Dades

Una vegada definides les entitats, ja es pot procedir a la transformació a un sistema de base de dades concret, passant a sentències específiques el nostre sistema de base de dades. En el cas d’InnovaCampus, el gestor de bases de dades escollit ha estat PostgreSQL.

Per més informació sobre el disseny de la base de dades consultar [4]

Breu introducció de PostgreSQL:

PostgreSQL és un servidor de base de dades relacional lliure, alliberat sota la llicència BSD. És una alternativa a altres sistemes de bases de dades de codi obert (com MySQL, Firebird y MaxDB), així com de sistemes propietaris com Oracle o DB2.

Algunes de les seves principals característiques són les següents:

- Claus foranes (foreign Keys).
- Disparadors (triggers).
- Vistes.
- Integritat transaccional.
- Accés concurrent multi-versió (no es bloquegen les taules, fins i tot, quan un procés escriu).
- Capacitat d’albergar programes en el servidor amb varis llenguatges.
- Herència de taules.
- Tipus de dades i operacions geomètriques.

Podeu trobar més informació sobre PostgreSQL a [18].

Capítol 3

Arquitectura de l'aplicació

El primer pas en el disseny d'una aplicació és decidir quina arquitectura s'utilitzarà. En aquest cas s'ha optat per una arquitectura de tres capes: capa de presentació, capa de negoci i capa persistència.

Es tracta d'una arquitectura freqüentment emprada en el marc de les aplicacions client-servidor com són les aplicacions web.

La finalitat bàsica d'aquesta arquitectura és proporcionar-nos un desacoblament total entre cadascuna de les capes, fet que s'aconsegueix utilitzant interfícies amb la finalitat que interactuïn les capes entre elles.

3.1 Introducció a l'arquitectura web

3.1.1 Arquitectura Client - Servidor

L'arquitectura client – servidor és un model per al desenvolupament de sistemes d'informació, en el qual les transaccions es divideixen en processos independents que cooperen entre sí per a intercanviar informació, serveis o recursos. Es denomina client al procés que inicia el diàleg o sol·licita els recursos i servidor, al procés que respon a les sol·licituds.

És el model d'interacció més comú entre aplicacions en una xarxa. Els principals components d'aquest esquema són els clients, els servidors i la infraestructura de comunicacions. En aquest model, les aplicacions es divideixen de forma que el servidor conté la part que ha de ser compartida per varis usuaris, i al client només s'hi manté la particular de cada usuari.

Podem sintetitzar-ho de la següent manera: El client és aquell que s'especialitza en fer peticions, és el que normalment tenim a la nostra computadora i aquell amb qui interactuem . Aquest però, no té les capacitats necessàries per a completar un servei (només fa peticions) , és aquí on entra el servidor, el qual es troba a l'altra banda de la connexió. L'únic que fa es esperar peticions i satisfer-les.

Això sí, de la mateixa manera que a la vida real les coses s'han de demanar correctament. Per a què client i servidor s'entenguin han de tenir un llenguatge en comú, un protocol (el protocol HTTP, el qual veurem en un apartat posterior, és el més utilitzat).

3.1.2 Aplicacions Web

En el punt anterior hem parlat de l'arquitectura client – servidor i n'hem sintetitzat les seves característiques. Presentem ara, les aplicacions Web com a cas particular d'aplicacions client – servidor. En els primers temps de la computació client – servidor, cada aplicació tenia el seu propi programa client i la seva interfície d'usuari, aquests havien de ser instal·lats separatament a cada estació de treball dels usuaris. Una millora al servidor, com a part de l'aplicació, requeria típicament una millora dels clients instal·lats a cadascuna de les estacions de treball, afegint un cost de suport tècnic i disminuint l'eficiència del personal.

En contrast, les aplicacions Web generen dinàmicament una sèrie de pàgines en un format estàndard, suportat per navegadors comuns com HTML o XHTML. S'utilitzen llenguatges interpretats pel client, com per exemple JavaScript, per tal d'afegir elements dinàmics a la interfície d'usuari. Generalment cada pàgina Web individual és enviada al client com un document estàtic, però la seqüència de pàgines prové d'una experiència interactiva.

Així doncs, podríem dir que en general (tot i que no necessàriament), el client accedeix al servidor mitjançant un navegador el qual li permet visualitzar les diferents pàgines Web .

3.1.3 HTTP: HyperText Transfer Protocol

El protocol més utilitzat per els clients i servidors de la web s'anomena HiperText Transfer Protocol (protocol de transferència d'hipertextos). Així com nosaltres en comuniquem utilitzant un idioma determinat, els servidors i clients de la World Wide Web utilitzen l'HTTP per a entendre's.

Aquest no és l'únic protocol utilitzat a la Web. Una de les característiques més importants de la Web, és que reuneix a tots els altres serveis “sota un mateix barret”. Però l'HTTP és el protocol primari, i es el que va ser específicament dissenyat per transferir documents o objectes hipermedials.

El protocol en sí defineix una transacció simple de quatre passes:

1. El client estableix una connexió amb el servidor.
2. El client fa una petició al servidor.
3. El servidor li torna una resposta amb el status i el contingut de la resposta.
4. S'acaba la connexió.

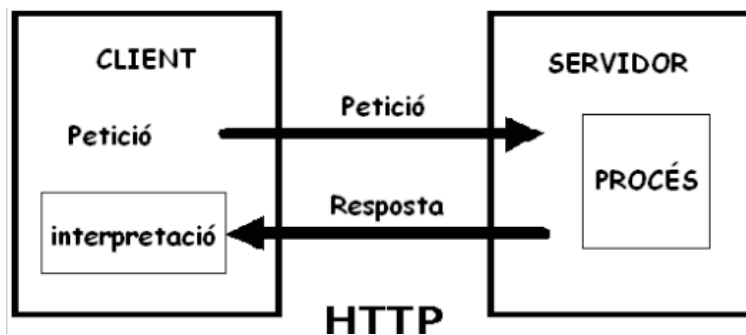


Figura 3.1: Arquitectura Client - Servidor

Propietats més rellevants:

- Es tracta d'un protocol sense estat (stateless). Això significa que no guarda informació sobre transaccions prèvies. Si bé això permet agilitzar el mecanisme, també es un inconvenient perquè moltes vegades es necessari manejar informació de passos previs.
- No proporciona dinamisme.
- Simplicitat. Permet que el servidor manegi poca càrrega per a cada petició, de manera que pot atendre més peticions simultàniament.
- Flexibilitat. Permet tipificar i transmetre qualsevol tipus de dades.
- Sense Connexió (*connectionless*). Pot haver-hi una única petició per connexió i després aquesta es talla, de manera que la utilització de recursos és la mínima necessària.

3.1.4 Aplicacions RIA: El nou paradigma en el desenvolupament d'aplicacions web.

El món del Web ha experimentat un creixement exponencial en els últims anys i en conseqüència han emergit un seguit de potents tecnologies que permeten desenvolupar aplicacions molt potents, les conegudes com aplicacions RIA.

Les aplicacions RIA estan enfocades a millorar la qualitat de les aplicacions web per tal d'assolir experiències d'usuari més pròximes a les que ofereixen les aplicacions d'escriptori.

Es podrien considerar com un nou tipus d'aplicacions amb més avantatges que les tradicionals aplicacions Web. Aquestes sorgeixen de la combinació de les aplicacions tradicionals i les aplicacions web. Per tant, es tracta d'aplicacions web que tenen les característiques i funcionalitats de les aplicacions tradicionals o d'escriptori.

Així doncs, podríem dir que les aplicacions RIA són aplicacions que:

- Ofereixen un ventall més ampli de components.
- La funcionalitat dels components és major, assimilant-se a una aplicació d'escriptori.
- S'eviten les recàrregues globals de la pàgina.
- Major suport a components multimèdia.
- Utilitzen un ampli ventall de tecnologies diferents:
 - AJAX
 - Flash
 - Flex
 - Portlets
 -

InnovaCampus, l'aplicació que dona lloc a la present memòria, quedaria englobada dins del marc d'aquest tipus d'aplicacions, és doncs, una aplicació RIA. En el transcurs de la memòria anirem fent èmfasi en les diferents tecnologies emprades en la construcció d'InnovaCampus.

3.2 El patró Model Vista Controlador

3.2.1 Introducció als Patrons de Disseny

Normalment es construeixen aplicacions Web en les que cada pàgina JSP gestiona serveis de seguretat, de recuperació de continguts i de navegació. Això ens porta a un model amb un alt cost de manteniment, en el que tenim grans quantitats de codi duplicat en nombroses pàgines (típica solució que es desenvolupa utilitzant la coneguda tècnica de copiar – empegar i modificar una mica).

Podem millorar enormement la qualitat d'aquestes aplicacions centralitzant i encapsulant alguns d'aquests mecanismes fent l'aplicació molt més mantenible, senzilla i neta, al eliminar gran quantitat de scriptlets (el codi java de les pàgines JSP). Per a la consecució d'aquests objectius no hi ha res millor que l'experiència condensada de molts anys de desenvolupament i disseny: els patrons de disseny.

Si ens remetem a una definició formal del que és un patró de disseny diríem que els patrons de disseny són descripcions d'objectes i classes comunicant-se, les quals són adaptades per a resoldre un problema de disseny general en un context particular.

El principal objectiu dels patrons de disseny és capturar bones pràctiques que ens permetin millor la qualitat del disseny de sistemes, determinant objectes que suportin rols útils dins d'un context específic, encapsulant complexitat, i fent-ho més flexible.

Com ja hem dit, els patrons de disseny contribueixen a donar flexibilitat i extensibilitat als nostres dissenys. Han demostrat ser una forma molt útil de reutilitzar dissenys, ja que no només nombren, abstraen i identifiquen aspectes claus d'estructures comunes de disseny, sinó que generalment són descrits en una forma específica documental, fent la seva comprensió i aplicació fàcil per al conjunt de programadors.

Així doncs podríem enumerar en els següents tres punts els aspectes més rellevants que ens aporten els patrons de disseny:

1. Contribueixen a reutilitzar dissenys, identificant aspectes claus de l'estructura d'un disseny que poden ser aplicats en una gran quantitat de situacions. La importància de la reutilització del disseny no és despreciable, ja que es proveeix de nombroses avantatges: redueix els esforços de desenvolupament i manteniment, millora la seguretat, eficiència i consistència dels nostres dissenys, i ens proporciona un considerable estalvi en la inversió.
2. Milloren la flexibilitat, modularitat i extensibilitat, factors interns i íntimament relacionats amb la qualitat percebuda per l'usuari.
3. Incrementen el nostre vocabulari de disseny, ajudant-nos a dissenyar des d'un major nivell d'abstracció.

3.2.2 El patró Model Vista Controlador (MVC)

El Model Vista Controlador (MVC) és un patró de disseny de software que separa les dades d'una aplicació, la interfície d'usuari, i la lògica de control en tres components diferents de forma que les modificacions al component de la vista poden ser realitzades amb un mínim impacte al component del model de dades. Això és útil ja que els models típicament tenen un cert grau d'estabilitat (depenent de l'estabilitat del domini del problema que està sent modelat), on el codi de la interfície d'usuari sigui més robust, degut a que el seu desenvolupador està menys propens a “trençar” el model mentre treballa de nou amb la vista.

El patró MVC és molt freqüent en el món de la Web, on la vista és la pàgina HTML i el codi que proveeix de dades dinàmiques a la pàgina.

Un termes generals, construir una aplicació utilitzant una arquitectura MVC implica definir tres classes de mòduls.

- **Model:** Aquesta és la representació específica del domini de la informació sobre el qual funciona l'aplicació. El model és una altra forma de cridar a la capa de domini. La lògica de domini afegeix significat a les dades.
- **Vista:** Aquest presenta el model en un format adequat per a interactuar, usualment un element de la interfície d'usuari.
- **Controlador:** Aquest respon als events, usualment accions de l'usuari i invoca canvis en el model i probablement a la vista.

En el següent esquema podem observar gràficament la idea de l'arquitectura MVC:

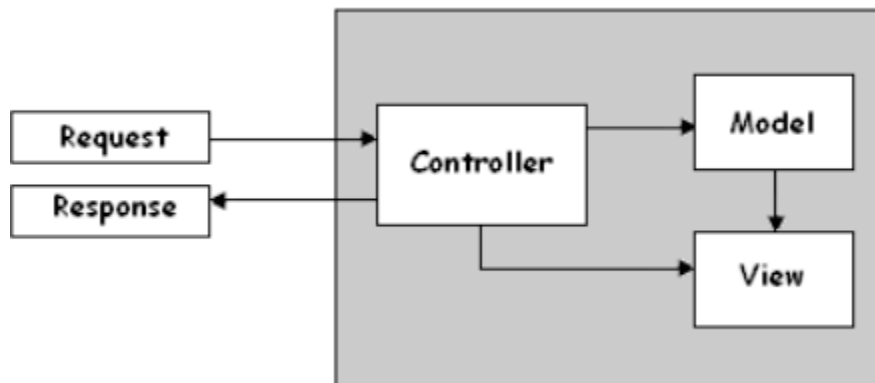


Figura 3.2: Arquitectura patró MVC

3.2.3 Introducció al framework SPRING MVC

El desenvolupament d'aplicacions software pot esdevenir força complex. El framework Spring MVC proporciona una solució elegant enfocada a la creació d'aplicacions en un entorn de treball Model Vista Controlador. Es tracta d'un framework amb una immensa implantació, gran acceptació, en continua renovació i d'una gran estabilitat. El nombre de característiques que ofereix és enorme i cobreix totes les capes que integren una aplicació. Spring no es limita tant sols a l'àmbit web i té multitud de capes d'integració amb un gran nombre de tecnologies.

Spring és un dels contenidors lleugers més populars. Els contenidors lleugers intenten proveir serveis per components que siguin el més senzills possibles. Així doncs, podríem citar com a alguns dels aspectes més interessants de Spring, la transparència dels objectes i l'existència de serveis nets i desconnectables. Tots aquets aspectes fan de Spring un Framework senzill i efectiu, amb les següents característiques:

- Implementació d'una solució MVC.
- Spring és a la vegada exhaustiu i modular. Implementa una arquitectura en capes molt consistent.
- Spring està creat des de els seus inicis per servir d'ajuda a la creació de codi fàcil de provar.
- Esdeveneix una important tecnologia d'integració. Integració amb diferents tecnologies: JSP/JSTL, Tiles, Velocity, FreeMarker, Excel, XSL.PDF, etc..
- Transparència.
- Possibilitats d'extensió.

Per obtenir més informació sobre Spring MVC consultar [Annex A].

3.3 Arquitectura InnovaCampus

Tal i com s'ha introduït anteriorment, en el cas d'InnovaCampus s'ha optat per una arquitectura de tres capes: capa de presentació, capa de negoci i capa persistència; amb la pretensió d'aconseguir un bon nivell de desacoblament entre les capes.

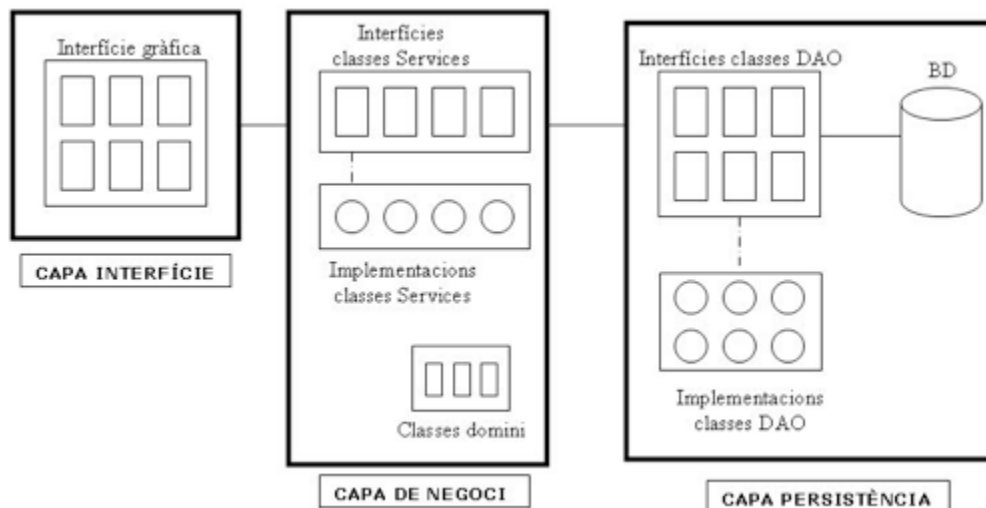


Figura 3.3: Arquitectura InnovaCampus de tres capes

La capa de negoci és la que s'encarrega de tota la funcionalitat de l'aplicació. Està formada per les classes de domini i els serveis de la lògica de negoci.

Tota la informació de l'aplicació s'emmagatzema en una base de dades relacional, ubicada en la capa de persistència, i per a ésser mapejada en el món dels objectes disposem d'interfícies DAO, amb les respectives implementacions. La capa de presentació o d'interfície és la que s'encarrega de generar les vistes de l'aplicació.

El redisseny de l'aplicació InnovaCampus s'ha fet utilitzant aquesta arquitectura de tres capes mitjançant la integració del Framework Spring amb JPA i Google Web Toolkit (GWT).

3.3.1 La capa de persistència

La capa de persistència d'una aplicació és l'encarregada d'emmagatzemar les dades i facilitar l'accés a les mateixes. Aquesta capa pot estar formada per un o més gestors de bases de dades que realitzen tot l'emmagatzemament de dades, reben les sol·licituds d'emmagatzemament i recuperació d'informació des de la capa de negoci.

En aquesta aplicació el gestor escollit ha estat PostgreSQL, i les classes que interactuen directament amb les dades són les classes DAO (Data Access Object) implementades en JPA (Java Persistence API). En el punt 2.6 s'ha especificat el disseny de la base de dades associada al model de persistència desenvolupat.

Per tal d'obtenir més informació sobre la implementació de la capa de negoci consultar [Annex B] i [4].

3.3.2 La capa de negoci

La capa de negoci és la que implementa la 'lògica de negoci' de l'aplicació. Aquesta 'lògica de negoci' correspon essencialment a les funcionalitats que l'aplicació ofereix als seus clients. Aquesta capa rep les sol·licituds de la interfície amb la finalitat de processar-les amb l'objectiu de retornar-ne posteriorment els resultats, en conseqüència també es comunica amb la capa de persistència per tal de desencadenar les corresponents transaccions d'informació.

Així doncs, la finalitat d'aquesta capa és la d'implementar totes les funcionalitats de l'aplicació, controlant la totalitat d'excepcions que poden donar-se i fent ús de les interfícies de les classes DAO (capa de persistència) per tal de gestionar la informació de la base de dades.

Per tal de efectuar un disseny elegant, tots els serveis creats per tal de donar resposta a la funcionalitat requerida per InnoVaCampus s'han agrupat en una sèrie de subsistemes en funció del perfil de les accions generades.

En el capítol 2, anàlisi de requisits, s'han mencionat totes les funcionalitats de l'aplicació, en funció de les diferents funcionalitats i per tal d'afavorir una implementació neta i clara del disseny s'han agrupat els serveis de la capa de negoci en quatre subsistemes diferents:

- **AdministrationManagement:** Aquest subsistema s'encarrega de tasques de caire més administratiu, com per exemple la gestió d'usuaris, la creació d'assignatures o la gestió d'anuncis.
- **SubjectManagement:** La finalitat bàsica d'aquest subsistema és la gestió del contingut de les assignatures. S'encarrega d'afegir, modificar i esborrar els temes i objectius de les diferents assignatures. Una assignatura està dividida en temes i a la vegada els temes estan dividits en objectius.
- **TestManagement:** La finalitat d'aquest subsistema és la de gestionar la creació de testos. És el responsable de la gestió de preguntes i respostes. Les preguntes formen part d'un objectiu en concret i cada pregunta té múltiples respostes. També s'encarrega d'associar recursos a preguntes i de la generació de tests.
- **StatisticsManagement:** Aquest subsistema és el que s'encarrega de gestionar la generació d'estadístiques i la consulta de tota la informació relacionada amb els resultats dels testos emmagatzemats.

Per tal d'obtenir més informació sobre la implementació de la capa de negoci consultar [Annex C] i [4].

3.3.3 La capa de presentació

3.3.3.1 Aplicacions d'Escriptori vs. Aplicacions Web

Podríem sintetitzar les característiques principals de les aplicacions tradicionals o d'escriptori de la següent manera:

- Les aplicacions d'escriptori ofereixen una interfície d'usuari rica en funcionalitat.
- Les eines de desenvolupament d'aplicacions d'escriptori ofereixen un conjunt de funcionalitats per facilitar la tasca al programador:
 - Conjunt de components predefinitos i personalitzables de forma declarativa en fase de disseny.
 - Plantilles de codi.
 - Entorn gràfic amigable amb funcionalitats visuals avançades.
 - Capacitat de creació de components propis que es poden incorporar a les aplicacions.

Pel que fa les aplicacions Web tradicionals, en sintetitzaríem les característiques principals de la següent forma:

- En general, les aplicacions web ofereixen una interfície d'usuari més aviat pobre.
- Una aplicació web està limitada per les tecnologies de base:
 - HTTP, un protocol que orienta la interacció de l'aplicació a un model (petició de recurs - resposta).
 - HTML, un format documental no orientat a components.
 - La connexió entre la interfície i l'aplicació es fa a través d'una aplicació externa, el navegador, sobre el qual el programador no té control.
- Major complexitat de desenvolupament: implica un nombre major de tecnologies diferents: HTML, CSS, Javascript, tecnologies de servidor...

Observant els avantatges i inconvenients d'ambdós tipus d'aplicacions, sembla lògic pensar, que l'evolució de les interfícies associades a les aplicacions Web serà la d'incorporar cada cop més funcionalitats associades a les aplicacions d'escriptori de tota la vida, precisament aquesta és la tendència que segueixen les anteriorment mencionades, aplicacions RIA, i això és el que s'ha fet amb InnovaCampus. Com ja hem indicat, l'aportació més important d'aquest treball ha estat el desenvolupament i integració d'aquesta capa de presentació.

3.3.3.2 La capa de presentació d'InnovaCampus

Tal i com hem introduït al llarg del capítol, per tal de realitzar la implementació de l'aplicació s'ha fet ús del framework Spring amb la finalitat de desenvolupar una aplicació MVC elegant i eficient.

El Model Vista Controlador (MVC) és un patró de disseny de software que separa les dades d'una aplicació, la interfície d'usuari, i la lògica de control en tres components diferents de forma que les modificacions al component de la vista poden ser realitzades amb un mínim impacte al component del model de dades.

La capa de presentació és la que fa referència a la capa d'interacció amb el client (navegador) de l'aplicació, per tant, fa referència tant a la interfície gràfica com a les crides que aquesta realitza al servidor. Per tal de implementar la capa de presentació d'InnovaCampus s'ha utilitzat el framework Google Web Toolkit, un seguit de llibreries i eines lliures que permeten implementar una aplicació web a partir de codi JAVA que un cop compilat proporciona els següents serveis:

- **HTML i JavaScript:** GWT compila el codi desenvolupat generant codi HTML i Javascript que es el que s'executa al navegador del client.
- **AJAX:** Capacitat que proporciona GWT a la interfície gràfica de l'aplicació.

La finalitat bàsica de l'elecció de GWT per tal de dur a terme la implementació de la capa de presentació ha estat la facilitat que aporta el framework per tal de generar aplicacions web AJAX complexes.

Al llarg del capítol 4 es detalla àmpliament el framework Google Web Toolkit i també s'explica la integració d'aquest framework amb Spring i JPA.

3.3.3.3 Aportacions d'AJAX a la capa presentació

El **JavaScript asincrònic i XML**, (*Asynchronous JavaScript and XML, AJAX, en anglès*), no és una tecnologia per si mateixa, sinó un terme que descriu una ònova aproximació davant de tecnologies ja existents com ara: HTML o XHTML, fulls d'estil en cascada, JavaScript, el model d'objectes de document, XML, XSLT, i l'objecte XMLHttpRequest. En combinar-les totes en el model d'AJAX, les aplicacions web poden fer actualitzacions ràpides i incrementals de la interfície d'usuari sense que calgui tornar a actualitzar la pàgina des del navegador. Això fa que l'aplicació respongui més ràpidament a les accions de l'usuari.

Com ja hem dit no es tracta d'una tecnologia nova, sinó que es tracta d'un nou concepte que relaciona les tecnologies web existents per millorar la usabilitat de la web.

Els objectius que assoleix són els següents:

- Millora de l'experiència d'usuari reduint els temps d'espera.
- Millora del Look & Feel de l'aplicació eliminant refrescos innecessaris.

Ajax proporciona interaccions dinàmiques de la web i interacció asíncrona amb el servidor.

Els següents diagrames mostren l'evolució dels models d'interacció d'usuari:

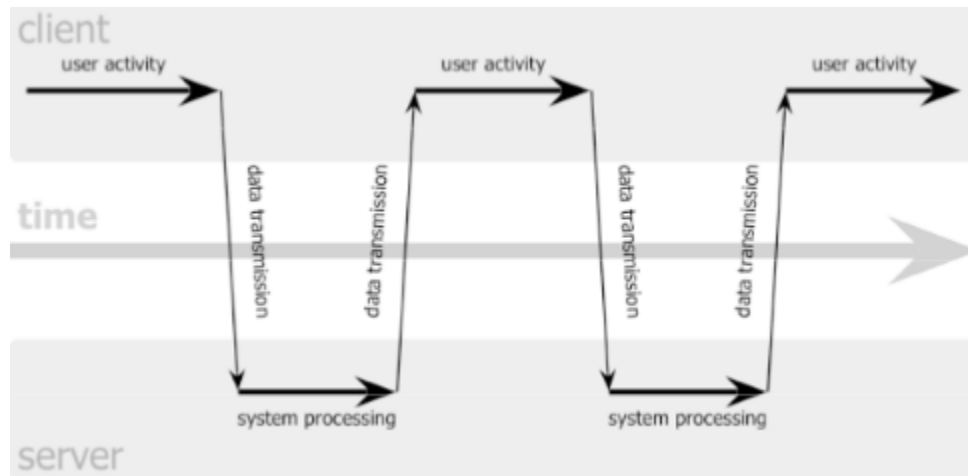


Figura 3.4: Model clàssic d'Aplicació Web (Comunicació sincron)

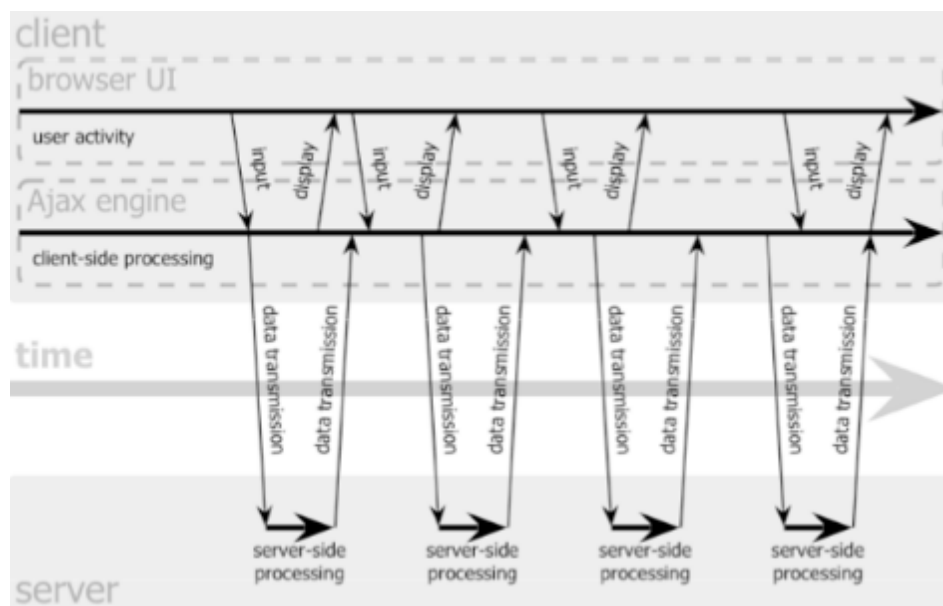


Figura 3.5: Model AJAX d'Aplicació Web (Comunicació asincron)

En els diagrames s'observa clarament les millores en quant a la interacció per part de l'usuari, ja que el sistema no requereix de refrescos innecessaris.

Podeu trobar més informació sobre AJAX a [Annex A] i [34]

Capítol 4

Disseny de la capa de presentació

La capa de presentació és la que fa referència a la capa d'interacció amb el client de l'aplicació, per tant, es refereix tant a la interfície gràfica com a les crides que aquesta realitza al servidor.

En el cas d'InnovaCampus la implementació de la capa de presentació s'ha dut a terme mitjançant la utilització del framework lliure Google Web Toolkit (GWT).

Al llarg d'aquest capítol es descriuran les característiques més importants de GWT tot detallant l'ús realitzat dins de la implementació d'InnovaCampus i s'especificarà el procediment d'integració d'aquest framework amb Spring i JPA.

4.1 Visió general de Google Web Toolkit (GWT)

GWT o Google Web Toolkit és un framework creat per Google que permet amagar la complexitat dels aspectes de la tecnologia AJAX. És compatible amb la majoria dels navegadors web. El concepte és força senzill, el desenvolupador crea codi Java que posteriorment serà compilat i transformat a HTML i JavaScript.

GWT ens aporta una API per a la capa presentació i un conjunt d'eines de desenvolupament que permeten implementar funcionalitats JavaScript d'alt rendiment dins del navegador del client.

4.1.1 Perquè GWT ?

Actualment, la creació d'aplicacions web és un procés pesat i propens a errors. Els desenvolupadors poden passar el 90% del seu temps estudiant les peculiaritats dels diferents navegadors. D'altra banda, la creació, la reutilització i el manteniment d'una gran quantitat de components AJAX i bases de codi JavaScript poden ser tasques complexes i delicades. Google Web Toolkit (GWT) facilita aquestes tasques tan feixugues oferint als desenvolupadors la possibilitat de crear i mantenir ràpidament aplicacions JavaScript amb interfícies complexes, però de gran rendiment, mitjançant el llenguatge de programació Java.

Un dels principals objectius de l'aplicació InnovaCampus és la d'oferir a l'usuari final una interactivitat total amb l'aplicació aplicant el concepte AJAX sempre que sigui possible. Com ja hem avançat,

GWT ofereix un seguit d'eines que ens permeten crear aplicacions AJAX d'una complexitat important. GWT aporta flexibilitat, simplicitat i rapidesa per tal d'obtenir aplicacions AJAX elegants, eficients i compatibles amb la majoria de navegadors actuals.

GWT permet obtenir aplicacions web similars a les aplicacions d'escriptori d'avui en dia, la potència d'aquest framework és tal que fins i tot es podria arribar a dubtar de si estem davant d'una aplicació web executada en un navegador o si estem davant d'una aplicació d'escriptori. A més a més, facilita la creació d'aplicacions mitjançant la utilització d'una sèrie de llibreries lliures que ens permeten crear interfícies gràfiques de forma similar a com ho fariem mitjançant les llibreries JAVA SWING o AWT.

Importància de la incorporació d'AJAX

Com ja hem avançat al llarg del capítol 3, AJAX són les sigles de Asynchronous Javascript And Xml, (JavaScript asíncron i XML), un conjunt de tecnologies que permeten actualitzar continguts web sense haver de tornar a carregar la pàgina. AJAX és asíncron en tant que les dades addicionals són demanades i carregades en un segon pla, sense interferir en la presentació i el comportament de la pàgina. Això obre la porta a una major interactivitat a les pàgines web, la qual cosa fa que l'aplicació web resultant s'assembli a una aplicació d'escriptori, millorant així l'experiència de l'usuari.

Plantejar una aplicació Web avui en dia sense fer ús d'AJAX seria una qüestió absurda, doncs AJAX aporta un dinamisme a les pàgines Web que millora l'interactivitat de l'aplicació millorant l'experiència d'usuari.

Així doncs, algunes de les característiques i aportacions més importants de la tecnologia AJAX serien:

- Aporta dinamisme i interactivitat a l'aplicació.
- Tecnologia basada en estàndards oberts com per exemple: XML, HTML, CSS, ...
- Millora la usabilitat de les aplicacions.
- Compatible amb qualsevol aplicació.
- Compatible amb altres tecnologies com per exemple FLASH.
- Adoptat per les principals empreses del sector.
- Independència de la tecnologia que s'utilitza al navegadors.
- Millora l'estètica del web.

4.1.2 Arquitectura GWT

GWT disposa de quatre components principals que en permeten el seu correcte funcionament: un compilador Java-a-JavaScript, un navegador web "hosted", i dos llibreries de classes creades íntegrament per Google:

Components de GWT:

- GWT Java-to-Javascript Compiler: Traductor de llenguatge JAVA a JavaScript.
- Hosted Web Browser: Executa l'aplicació en JAVA tot utilitzant la VM.
- JRE Emulation Library: Conte les biblioteques més importats.
- GWT Web UI Class Library: Ens permet la creació d'objectes com caixes de text, botons, imatges, ...



Figura 4.1: Eines de GWT

Algunes de les característiques més importants que ens aporta GWT són una fàcil comunicació (RPC) entre navegador i servidor, la possibilitat de depuració en temps real del codi, compatibilitat total amb molts navegadors, integració amb Junit, internacionalització, la utilització d'una eina de codi lliure, entre altres.

Modalitats d'execució:

GWT ens proporciona dues modalitats d'execució:

1. **Mode Host:** L'aplicació s'executa en codi bytecode de Java dins la Maquina Virtual per tal de facilitar la feina als desenvolupadors.
2. **Mode Web:** L'aplicació s'executa com a codi JavaScript i HTML.

4.2 Desenvolupament d'aplicacions RIA mitjançant GWT

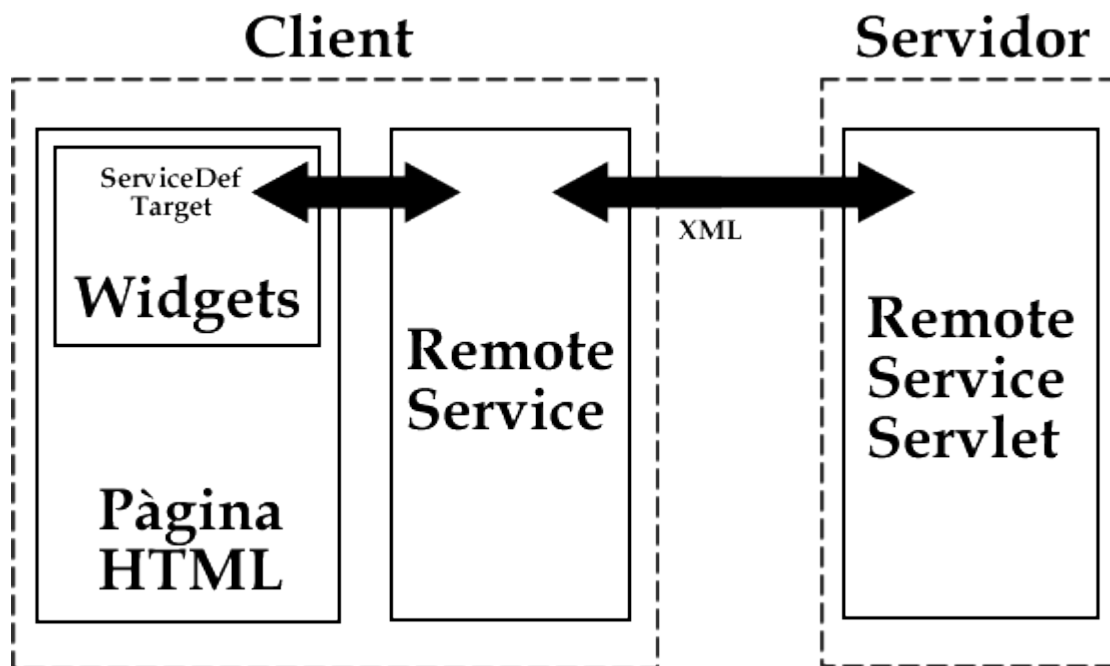


Figura 4.2: Arquitectura Client / Servidor amb GWT

Tal com hem dit, GWT ens aporta una sèrie d'eines per tal de facilitar el desenvolupament de la interfície i la comunicació amb el servidor, d'aplicacions web AJAX. L'esquema de la figura 4.2 ens descriu les parts que conformen una aplicació web desenvolupada amb GWT.

1. La interfície d'usuari amb tots els widgets bàsics, panels, events, ...
2. Els remote services que ens han de permetre connectar amb el servidor.
3. Els remote service servlet que han de processar les dades enviades per remote services i retornar un resultat al remote service del client.

Anem doncs a descriure cada una de les seves parts posant exemples.

4.2.1 Desenvolupament de la interfície d'usuari

GWT ens permet utilitzar un conjunt de components d'interfície d'usuari anomenats Widgets bàsics (textarea, textbox, buttons, menus, ...), per tal de poder crear aplicacions AJAX o altres Widgets.

Hem de tenir en compte que un Widget és una eina virtual interactiva que ens ofereix un servei, en aquest cas GWT ens ofereix un ventall d'eines per tal de poder desenvolupar aplicacions però nosaltres també podem generar nous widgets a partir dels que ens ofereix GWT i utilitzar-los en les nostres aplicacions web, per tant tot formulari, llista, menú, indicador, ... que hem creat per InnovaCampus i que ens aporta un servei l'anomenarem també Widget.

Igual que en la majoria dels llenguatges de programació, els components de la UI (Widgets bàsics) son agrupats en panells que determinen la ubicació dels mateixos dins la pantalla. A continuació veurem un exemple real del Widget de modificació de contrasenya de l'aplicació InnovaCampus.

```
public class ChangePassword extends VerticalPanel {

    public ChangePassword() {

        setWidth("100%");
        setStyleName("HPForms");

        HorizontalPanel HP2 = new HorizontalPanel();
        HorizontalPanel HP3 = new HorizontalPanel();
        HorizontalPanel HP4 = new HorizontalPanel();
        HorizontalPanel HP5 = new HorizontalPanel();
        HP2.setStyleName("HPanelForm");
        HP3.setStyleName("HPanelForm");
        HP4.setStyleName("HPanelForm");
        HP5.setStyleName("HPanelForm");

        Label loldp = new Label(new IText().getText().OldPassword());
        Label lnewp = new Label(new IText().getText().NewPassword());
        Label lrnewp = new Label(new IText().getText().RepeatNewPassword());
        loldp.setStyleName("LabelForm");
        lnewp.setStyleName("LabelForm");
        lrnewp.setStyleName("LabelForm");

        final PasswordTextBox toldp = new PasswordTextBox();
        toldp.setName("oldpassword");
        toldp.setStyleName("TextBox");
        final PasswordTextBox tnewp = new PasswordTextBox();
        tnewp.setName("newpassword");
        tnewp.setStyleName("TextBox");
        final PasswordTextBox trnewp = new PasswordTextBox();
        trnewp.setName("rnewpassword");
        trnewp.setStyleName("TextBox");

        final FormError Error = new FormError();
        Error.setTextLoading(new IText().getText().Loading1());
        Error.setVisible(false);

        Button BTSettingProfile = new Button(new IText().getText().ChangePassw());

        HP2.setWidth("100%");
        HP2.add(loldp);
        HP2.add(toldp);
        HP2.setCellHorizontalAlignment(loldp, ALIGN_RIGHT);
        HP2.setCellHorizontalAlignment(toldp, ALIGN_LEFT);
        HP2.setCellWidth(loldp, "30%");
        HP2.setCellWidth(toldp, "70%");

        HP3.setWidth("100%");
        HP3.add(lnewp);
        HP3.add(tnewp);
        HP3.setCellHorizontalAlignment(lnewp, ALIGN_RIGHT);
        HP3.setCellHorizontalAlignment(tnewp, ALIGN_LEFT);
        HP3.setCellWidth(lnewp, "30%");
        HP3.setCellWidth(tnewp, "70%");

        HP4.setWidth("100%");
        HP4.add(lrnewp);
        HP4.add(trnewp);
        HP4.setCellHorizontalAlignment(lrnewp, ALIGN_RIGHT);
        HP4.setCellHorizontalAlignment(trnewp, ALIGN_LEFT);
        HP4.setCellWidth(lrnewp, "30%");
        HP4.setCellWidth(trnewp, "70%");

        HP5.setWidth("100%");
        HP5.add(BTSettingProfile);
        HP5.setCellHorizontalAlignment(BTSettingProfile, ALIGN_LEFT);
        HP5.setCellWidth(space, "30%");
        HP5.setCellWidth(BTSettingProfile, "70%");

        add(Error);
        setCellHorizontalAlignment(Error, ALIGN_CENTER);
        add(HP2);
        add(HP3);
        add(HP4);
        add(HP5);
    }
}
```

Figura 4.3: Codi GWT del widget de modificació de contrasenya

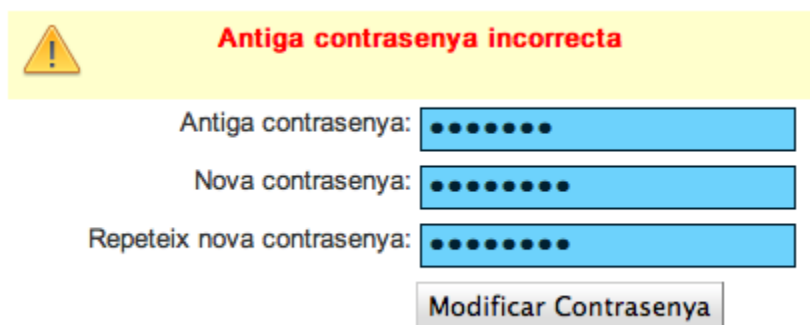
Com es pot observar a la figura 4.2 la classe ChangePassword hereda de la classe VerticalPanel definida per GWT i que fa referència a una taula HTML.

Les primeres línies ens configuren les característiques d'aquesta taula com són l'amplada i l'etiqueta CSS associada, a continuació definim 5 noves files on també els hi associem a cadascuna l'etiqueta CSS corresponent. just després definim els labels dels tres textbox que apareixeran al formulari (*loldp*, *lnewp*, *lrnewp*) i els textbox que contindran l'antiga i la nova clau (*toldp*, *tnewp*, *trnewp*).

També definirem un panel informatiu dels errors que es generin i un botó per acceptar la sol·licitud de canvi de contrasenya.

Finalment afegirem tots els elements a cada una de les files creades anteriorment (HP2, HP3, HP4, HP5) i introduïrem les files dins la taula.

El resultat el podem veure a la figura 4.3.



The image shows a web form for changing a password. At the top, there is a yellow rectangular box containing a warning icon (a triangle with an exclamation mark) and the text "Antiga contrasenya incorrecta" in red. Below this box, there are three input fields, each preceded by a label: "Antiga contrasenya:", "Nova contrasenya:", and "Repeteix nova contrasenya:". Each input field is a blue rectangle with black dots representing masked text. At the bottom of the form is a button with the text "Modificar Contrasenya".

Figura 4.4: Aspecte del widget de modificació de contrasenya

4.2.2 Comunicació dels widgets amb el servidor

Una diferència fonamental entre Google Web Toolkit i les tradicionals aplicacions web, és que les aplicacions GWT no necessiten d'altres pàgines web mentre són executades.

Com que les pàgines construïdes amb GWT funcionen com aplicacions sobre el navegador, aquestes no necessiten fer noves peticions al servidor per a, per exemple, realitzar actualitzacions de la interfície d'usuari. No obstant això, com totes les aplicacions client/servidor, els programes en GWT necessitaran demanar certes dades al servidor per a realitzar determinades tasques.

El mecanisme per interactuar amb el servidor a través de la xarxa és l'anomenat Remote Procedure Call (RPC), que vindria a ser una crida a procediments remots.

4.2.2.1 RPC Invocació a procediments remots

El concepte de RPC és molt similar al mecanisme ofert per JAVA per invocar un mètode de forma remota RMI (Java Remote Method Invocation). Una diferència fonamental entre GWT i les tradicionals aplicacions web, és que les aplicacions GWT no necessiten d'altres pàgines web mentre són executades. Tenint en compte que les pàgines desenvolupades amb GWT funcionen com aplicacions sobre el navegador, aquestes no necessiten fer noves peticions al servidor per realitzar actualitzacions de la interfície d'usuari. Tanmateix com totes les aplicacions client/servidor, els programes amb GWT necessiten demanar certes dades al servidor per tal de determinar certes tasques.

El mecanisme per interactuar amb el servidor a través de la xarxa és el conegut com a "remote procedure call" o "crida a procediment remot". El RPC amb GWT permet al client enviar i rebre objectes de Java sobre HTTP.

Quan és utilitzat adequadament, RPC dóna l'oportunitat de moure tota la lògica de la interfície d'usuari al client, cosa que millora el funcionament de l'aplicació, redueix l'ample de banda a utilitzar, redueix la càrrega al servidor i li presenta a l'usuari una experiència molt més agradable que les aplicacions web amb tecnologia tradicional.

El codi del servidor que és invocat des del client és freqüentment anomenat servei.

Cadascun dels serveis disposa d'una petita família d'interfícies i classes. Algunes d'aquestes classes, com el servei PROXY, són generades automàticament i pràcticament invisibles a nosaltres. El patró de classes és idèntic per tots els serveis que s'implementin, per la qual cosa, és bo familiaritzar-se una mica amb els termes i propòsits de cadascuna de les capes.

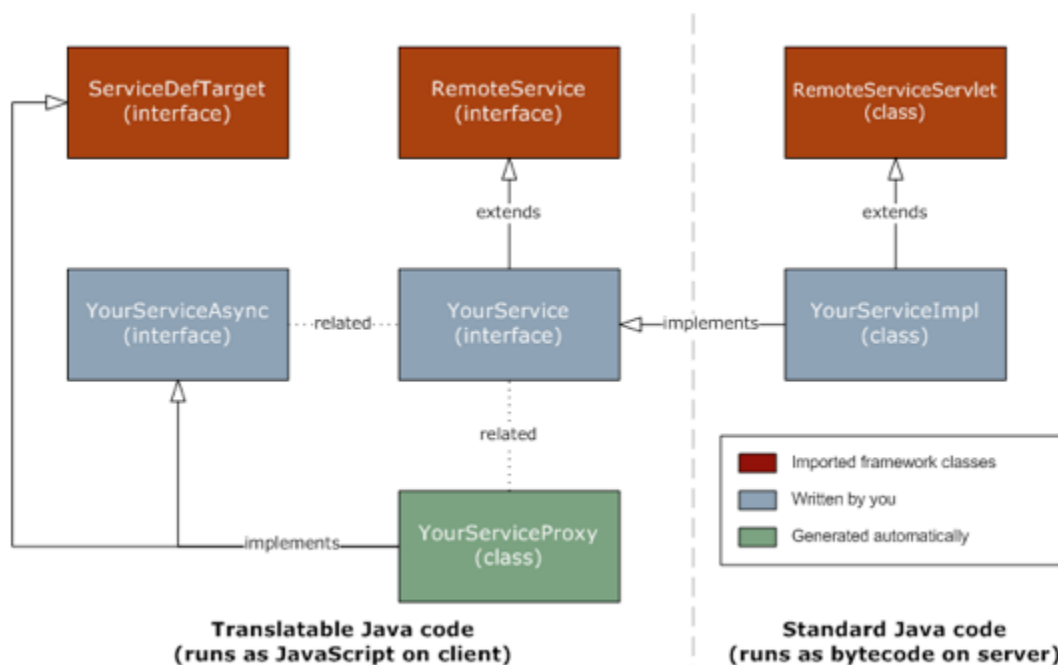


Figura 4.5: RPC - Components de la invocació de procediments remots

4.2.2.2 Creació d'un servei RPC mitjançant GWT

Per tal de poder crear l'estructura de la figura 4.5, hem de crear una interfície (síncrona) que hereti de RemoteService on hi haurà la declaració de tots els mètodes del servei.

```

public interface Servi extends RemoteService {
    public String Metode1(String valor);
    public LinkedList Metode2(int valor1, int valor2);
    public int Metode3();
}
  
```

Figura 4.6: RemoteService

La implementació d'aquesta interfície es realitza al costat del servidor la qual heretarà de RemoteServiceServlet. De fet és com un servlet que hereta de RemoteServiceServlet ja que aquest es l'encarregat de serialitzar els mètodes de forma automàtica.

```

public class ServeiImpl extends RemoteServiceServlet implements Servei {

    public String Metode1(String valor){
        // Accions a realitzar
    }

    public LinkedList Metode2(int valor1, int valor2){
        // Accions a realitzar
    }

    public int Metode3(){
        // Accions a realitzar
    }
}

```

Figura 4.7: RemoteServiceServlet

Abans de poder realitzar invocacions remotes des del client, s'ha de crear una altra interfície (asíncrona) basada en la interfície original (síncrona).

```

public interface ServeiAsync {
    public void Metode1(String valor, AsyncCallback callback);
    public void Metode2(int v1, int v2, AsyncCallback callback);
    public void Metode3(AsyncCallback callback);
}

```

Figura 4.8: RemoteService Asincron

La invocació de mètodes asíncrons requereix que el realitzador de la invocació, passi un objecte callback per paràmetre ja que la informació de retorn es realitza per mitja d'aquest objecte. Per tant la forma de declaració de les interfícies d'usuari ha de seguir escrupolosament el format següent:

- Nom_del_servei (síncrona).
 - Mètodes definits de forma normal.
- Nom_del_serveiAsync (asíncrona).
 - Tots els mètodes retornaran void.
 - Tots el mètodes incorporaran un nou paràmetre AsyncCallback.

Finalment aquest serà mapejat com un servlet al document 'web.xml'

4.2.2.3 Crides a procediments remots RPC

Un cop tenim el servei RPC desenvolupat és hora de cridar-lo des de la part de la interfície. Anem doncs a veure com es realitza a partir de l'exemple del punt 4.2.1 .

```
BTSettingProfile.addClickHandler(new ClickHandler() {

    public void onClick(ClickEvent arg0) {
        Error.setVisible(true);
        final RemoteServicesAsync srv = GWT.create(RemoteServices.class);
        srv.changePassword(User.getNIF(), tnewp.getText(), new AsyncCallback<String>() {

            public void onSuccess(String resultText) {
                Error.setTextSuccess(resultText);
            }

            public void onFailure(Throwable result) {
                Error.setTextError(resultText);
            }
        });
    }
});
```

Figura 4.9: Crida al procediment remot de canvi contrasenya

Com podem observar hem afegit al boto creat a l'exemple anterior el control de l'event de clic, per tant quan un usuari premi el boto s'executarà el codi contingut dins del mètode onClick.

Primerament activem la visualització del camp del panel d'error que en principi estava ocult, a continuació realitzem la crida al servei procediment remot changePassword per tal de ens modifiqui la contrasenya i ens retorni un missatge de que tot ha anat bé, si hem obtingut resposta per part del servidor s'executarà el mètode onSuccess i es presentarà per pantalla el missatge de que tot ha anat bé, en cas contrari ens retornarà l'error que s'ha produït i ens el presentarà per pantalla.

4.3 La serialització

Els objectes que es transfereixen mitjançant els serveis GWT entre el client i el servidor han de ser codificats en un format llegible per tal d'enviar-los a través de la xarxa.

En aquest cas GWT utilitza XML per tal de transferir les dades entre el client i el servidor. L'encarregat de realitzar la transformació automàtica dels objectes a XML és la classe `RemoteServiceServlet`.

4.3.1 Tipus de variables serialitzables

Un dels requisits de Google Web Toolkit és que els paràmetres i les variables de retorn siguin serialitzables, la qual cosa significa que han de complir amb certes restriccions. Google Web Toolkit intenta que la serialització sigui el més simple possible, de manera que, seguint adequadament les regles de serialització definides, l'ús n'esdevé força intuïtiu.

Un tipus de variable pot ser serialitzable i emprat en una interfície de servei si:

- És primitiu com ara `char`, `byte`, `short`, `int`, `long`, `boolean`, `float`, o `double`.
- És `String`, `Date`, o un embolcall primitiu com per exemple `Character`, `Byte`, `Short`, `Integer`, `Long`, `Boolean`, `Float`, o `Double`.
- És un arranjament de variables serialitzables (incloent altres arrays serialitzables).
- És una classe serialitzable definida per l'usuari. té almenys una subclasse serialitzable.

4.3.2 Classes serialitzables definides per l'usuari

Una classe definida per l'usuari és serialitzable si:

- És assignable a `IsSerializable` o `Serializable`, ja sigui perquè directament implementa una d'aquestes interfícies o perquè deriva des d'una superclasse d'aquestes.
- Totes les instàncies de camps no final, i no transient són serialitzables.
- Té un constructor públic per defecte (sense arguments).

La paraula reservada `transient` és en certa manera respectada, així que els valors en camps transient no són intercanviats durant crides a procediments remots. Els camps que són declarats final tampoc són intercanviats durant les crides a procediments remots, així que generalment també han de ser marcats com a transient.

4.3.3 Polimorfisme

Les crides a procediments remots de Google Web Toolkit suporten paràmetres i tipus de retorn polimòrfics. No obstant, amb la finalitat de fer un ús òptim del polimorfisme, cal que siguem molt específics en el disseny, sobretot durant la definició de les interfícies de servei.

4.3.4 Tipus d'arguments

Les classes associades a col·leccions com per exemple `java.util.Set` i `java.util.List` tenen un funcionament especial i delicat, ja que operen en termes d'instàncies de `Object`. Per fer col·leccions serialitzables, cal especificar el tipus particular d'objectes esperats per ser continguts. Això requereix utilitzar la notació `@gwt.typeArgs`, una anotació especial de Javadoc. Definir el tipus d'objecte `Collection` restringeix que la col·lecció només contingui objectes d'aquest tipus, cosa necessària perquè el generador de codi JavaScript de Google Web Toolkit pugui crear codi eficient. Afegir un objecte a una col·lecció que violi aquesta regla faria que l'aplicació es comportés de manera indefinida.

El següent exemple mostra com afegir camps de tipus `Collection` a una classe serialitzable.

```
public class LaMevaClasse implements IsSerializable {
    /**
     * Objecte set que sempre contindrà cadenes.
     *
     * @gwt.typeArgs
     */
    public Set setOfStrings;

    /**
     * Objecte Map que sempre contindrà dos cadenes
     * String del tipus clau - valor
     *
     * @gwt.typeArgs
     */
    public Map mapOfStringToString;

    /**
     * Constructor per defecte. Es tracta d'un requisit
     * per a que la classe sigui serialitzable.
     */
    public LaMevaClasse() {}
}
```

Figura 4.10: Arguments per a la serialització

Observem que no hi ha necessitat d'especificar el nom de l'objecte a la declaració `@gwt.typeArgs`. D'una forma similar, per tal d'incloure paràmetres i tipus de retorn ho fariem de la següent manera:

```
public interface ElMeuServei extends RemoteService {  
    /**  
     * La primera anotación indica que el parámetro llamado 'c' es un  
     * La primera anotació indica que el paràmetre anomenat 'c' és un  
     * objecte List que únicament conté objectes Integer. La segona  
     * anotació indica que l'objecte List retornat únicament conté  
     * objectes String (s'observa que no cal especificar el nom en els  
     * valors de retorn.  
     *  
     * @gwt.typeArgs c <java.lang.integer>  
     * @gwt.typeArgs <java.lang.string>  
     */  
    List reverseListAndConvertToStrings(List c);  
}
```

Figura 4.11: Arguments per la serialització

Als exemples hem pogut observar que l'anotació dels paràmetres ha d'incloure el nom del paràmetre en qüestió, a més a més del tipus d'objecte de la col·lecció, en canvi això no és necessari en les anotacions de valors de retorn.

Podeu trobar més informació sobre Serialització a [11]

4.4 Integració de GWT amb Spring Framework

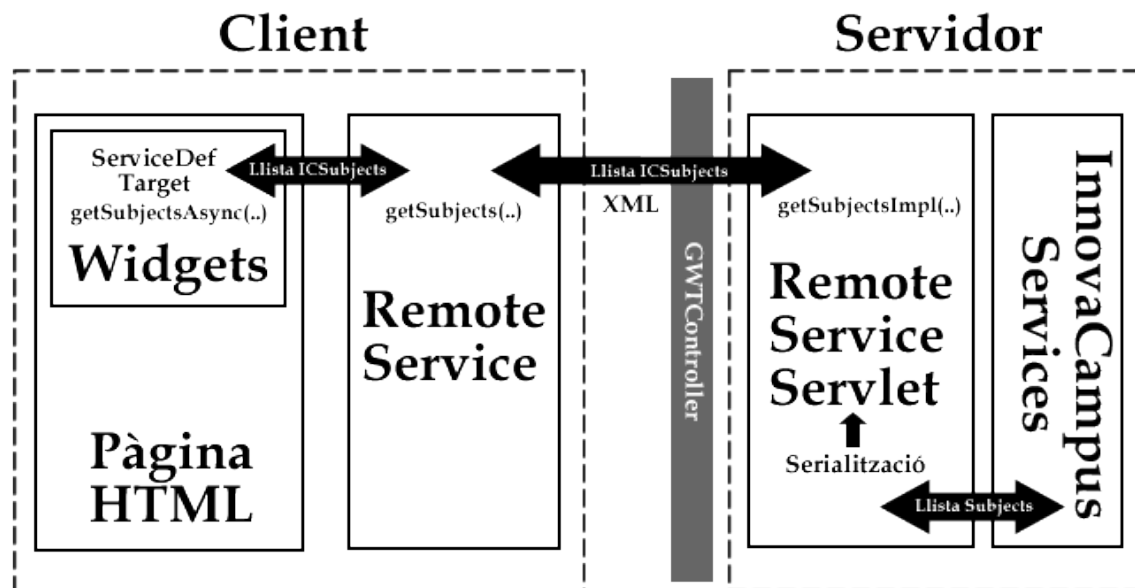


Figura 4.12: Arquitectura d'InnovaCampus amb GWT

Com ja s'ha explicat anteriorment, InnovaCampus ha estat desenvolupat mitjançant el Framework Spring per tal de separar l'aplicació en tres capes (Presentació, Negoci i Persistència) i optimitzar-ho al màxim.

El nostre propòsit ha consistit en redissenyar tota la capa presentació mitjançant tecnologies AJAX per tal d'aconseguir interactivitat i major velocitat. Per aquest motiu es va decidir d'utilitzar GWT ja que ens aporta rapidesa i claretat en el procés de desenvolupament d'aquesta nova capa presentació amb tecnologies AJAX.

Una de les tasques més importants del nostre treball ha estat, el procés d'integració de l'arquitectura d'una aplicació GWT (explicada en el punt 4.2) amb l'arquitectura d'InnovaCampus. La figura 4.12 ens descriu de forma gràfica, la nova arquitectura obtinguda després del procés d'integració.

Podeu trobar més informació sobre integració de GWT amb Spring a [35] i [36].

4.4.1 Com funciona Spring Framework ?

Spring és un framework de codi obert de desenvolupament d'aplicacions que ens permet implementar de forma fàcil i àgil el patró MVC (Model Vista Controlador).

Una de les millores que aportades per Spring és la implementació del patró MVC. Per tal de poder fer això Spring disposa d'un servlet que realitza la tasca de Front Controller o controlador. Això significa que cada una de les peticions o requests que generi l'usuari, passaran a través d'aquest servlet. El nom que rep aquest servlet és Dispatcher-Servlet el qual delegarà el processament del request rebut a un altre component.

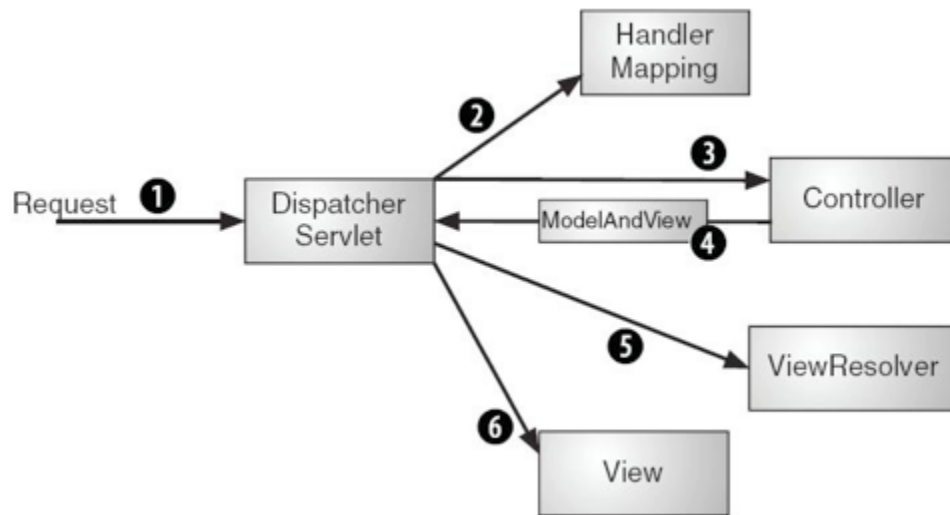


Figura 4.13: MVC mitjançant Spring Framework

El Handler Mapping té l'objectiu d'indicar al Dispatcher quin serà el controlador que ha de gestionar el request que ha enviat l'usuari.

Per poder implementar un controlador sobre Spring, és necessari crear una classe que hereti de Controller.

Un cop rebut el request al controlador, dins d'aquest es construeix un objecte ModelAndView. La funció principal d'aquest objecte es, inserir les dades resultants obtingudes de diferents serveis de la capa negoci al model, per tal de que el dispatcher els desplegui a la vista.

Per tant el dispatcher rep la resposta del controlador en forma d'un objecte ModelAndView i aquest desplega la vista a partir de les dades enviades pel controlador mitjançant aquest objecte.

4.4.2 Creació d'un exemple simple amb Spring Framework

Primerament hem de mapejar al nostre web.xml el Dispatcher-Servlet, per tal de fer que totes les crides passin primerament per aquest servlet. Això ho fem de la següent manera:

```
<servlet>
  <servlet-name> dispatcher </servlet-name>
  <servlet-class> org.springframework...DispatcherServlet </servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name> dispatcher </servlet-name>
  <url-pattern> *.htm </url-pattern>
</servlet-mapping>
```

Figura 4.14: Mapeig del web.xml

Per tant amb aquestes línies totes les peticions 'htm' seran delegades al Dispatcher-Servlet.

Anem ara a definir el HandlerMapping per tal d'associar peticions amb controlador, gràcies a això aconseguirem que el dispatcher sigui capaç de delegar cada petició rebuda, al seu controlador corresponent.

En aquest cas, es mapeja controlador a URL basats en el nom del bean.

```
<bean id="urlMapping" class="org.springframework...UrlHandlerMapping">
  <property name="urlMap">
    <map>
      <entry key="home.htm"><ref bean="home"/></entry>
      ...
    </map>
  </property>
</bean>
<bean id="home" class="org.elprojecte.server.controllers.homeController"/>
```

Figura 4.15: Mapeig del Dispatcher-Servlet

Amb això aconseguim que totes les peticions que facin els nostres clients a 'home.htm' siguin delegades al homeController.

```
public class HomeController extends AbstractController {

  public ModelAndView handlerRequestInternal(HttpServletRequest request ...)
  {
    ModelAndView mav = new ModelAndView("home");
    return mav;
  }
}
```

Figura 4.16: El Controlador HomeController

4.4.3 Integració dels serveis GWT amb Spring

Per tal de poder integrar els serveis GWT amb l'estructura d'un projecte Spring, primerament hem d'aconseguir redireccionar les crides RPC dels clients al Dispatcher-Servlet de Spring per tal de que sigui aquest, qui faci la funció de delegar al controlador pertinent les crides rebudes.

Això ho aconseguim mapejant els serveis GWT al Dispatcher-Servlet tot delegant la feina, al controlador corresponent que en aquest cas serà un controlador general per a tots els serveis RPC.

Iniciem redireccionant totes les peticions RPC al Dispatcher-Servlet dels RPC's.

```
<servlet>
  <servlet-name> dispatcherRPC </servlet-name>
  <servlet-class> org.springframework...DispatcherServlet </ servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name> dispatcherRPC </servlet-name>
  <url-pattern> *.rpc </url-pattern>
</servlet-mapping>
```

Figura 4.17: Redireccionament peticions rpc

A continuació mapegem totes les peticions amb el controlador GWTController.

```
<bean class="org.springframework...UrlHandlerMapping">
  <property name="mappings">
    <value>
      /**/Servei1.rpc=Control1
      /**/Servei2.rpc=Control2
      ...
    </value>
  </property>
</bean>

<bean name="Control1" class="projecte.server.controllers.GWTController">
  <property name="remoteService">
    <bean class="projecte.server.rpc.Servei1Impl"/>
  </property>
</bean>

<bean name="Control2" class="projecte.server.controllers.GWTController">
  <property name="remoteService">
    <bean class="projecte.server.rpc.Servei2Impl"/>
  </property>
</bean>
```

Figura 4.18: Mapeig peticions controlador GWT

Amb això haurem aconseguit delegar les peticions rebudes al Dispatcher-Servlet als serveis GWT corresponents per mitja del controlador GWTController.

Que fa GWTController ?

Tal com hem pogut veure anteriorment, al mapeig de GWTController afegim una propietat `RemoteService` que apunta al servei GWT corresponent a la crida RPC inicial. Per tant quan un client realitza una petició mitjançant RPC, aquesta és redirigida al `DispatcherServlet` el qual delega la petició a `GWTController` i aquest la envia de forma indirecta, al servei GWT corresponent.

A diferència dels controladors Spring tradicionals, aquest controlador no retorna cap objecte `ModelAndView`, sinó que la resposta és enviada al client, mitjançant el canal tradicional en el que funcionen els serveis GWT. De fet amb GWT la vista ja es troba totalment separada de la lògica de processament i per tant retornar un objecte `ModelAndView` no tindria gaire sentit, ja que els serveis GWT sols envien i reben dades específiques no vistes.

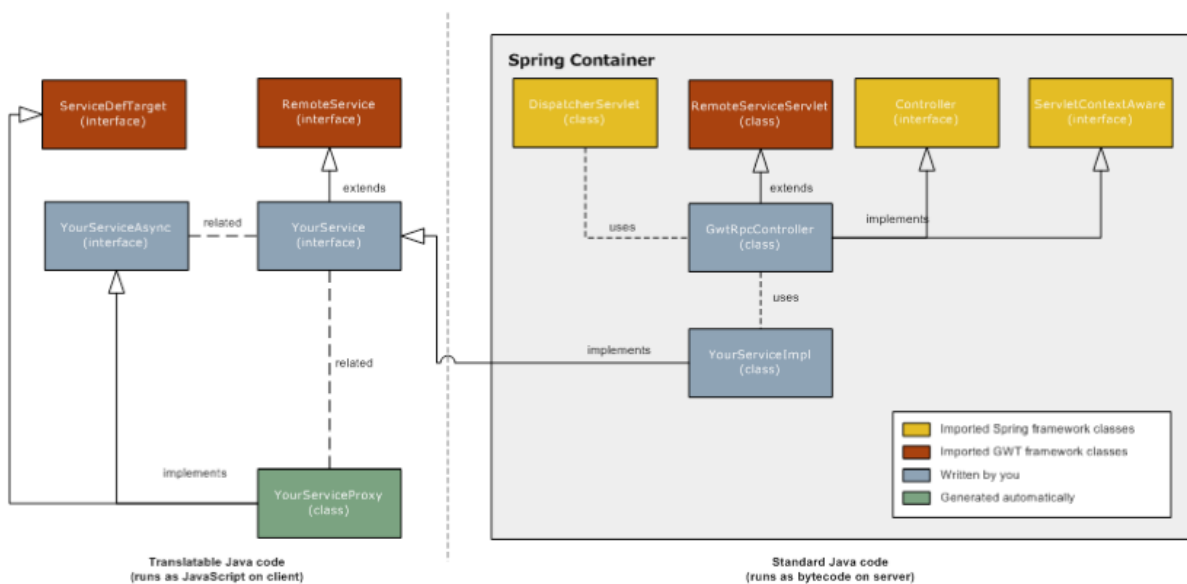


Figura 4.19: Diagrama integració GWT + Spring

Amb això aconseguim tenir integrat GWT i Spring per tal de que qualsevol tipus de petició sigui rebuda per un `DispatcherServlet` i processada per un `Controller`.

Com treballa GWTController ?

Anem a veure el codi de GWTController.

```
public class GWTController extends RemoteServiceServlet implements Controller,ServletContextAware {

    private static final long serialVersionUID = 1L;
    private ServletContext servletContext;
    private RemoteService remoteService;
    private Class<? extends RemoteService> remoteServiceClass;

    public ModelAndView handleRequest(HttpServletRequest request, HttpServletResponse response) throws Exception {
        doPost(request, response);
        return null;
    }

    @Override
    public String processCall(String payload) throws SerializationException {
        try {
            RPCRequest rpcRequest = RPC.decodeRequest(payload, this.remoteServiceClass);
            return RPC.invokeAndEncodeResponse(remoteService, rpcRequest.getMethod(), rpcRequest.getParam(), rpcRequest.getSerialization());
        } catch (IncompatibleRemoteServiceException ex) {
            getServletContext().log("An IncompatibleRemoteServiceException was thrown while processing this call.", ex);
            return RPC.encodeResponseForFailure(null, ex);
        }
    }

    @Override
    public ServletContext getServletContext() {
        return servletContext;
    }

    public void setServletContext(ServletContext servletContext) {
        this.servletContext = servletContext;
    }

    public void setRemoteService(RemoteService remoteService) {
        this.remoteService = remoteService;
        this.remoteServiceClass = this.remoteService.getClass();
    }
}
```

Figura 4.20: Codi GWTController

Bàsicament ens centrarem en els mètodes `handleRequest` i `processCall`. Inicialment quan el `DispatcherServlet` delega la feina al controlador `GWTController`, `handleRequest` és el mètode que rep la petició. Aquest genera un `doPost` que posteriorment crida el mètode `processCall`, el qual invoca el `remoteService` que va associat amb el `GWTController` o sigui el servei GWT que hem mapejat anteriorment.

Tot i que mitjançant aquest mètode podem tenir integrat totalment GWT amb Spring, en quan ens posem a treballar ens adonem que el controlador i el servei GWT estan totalment incomunicats i que aquests no es poden passar dades entre ells.

Comunicar GWTController amb els serveis GWT

La idea trivial seria accedir a dades de sessió des dels serveis GWT però el problema el trobem quan ens adonem de que els serveis GWT no tenen accés al request dels controladors i per tant al request de la petició. Tenint en compte que des de GWT no tenim accés a l'objecte `HTTPSession`, ens veiem obligats a enviar aquesta informació al servei GWT des del controlador.

Comunicació indirecta

Consisteix en crear un controlador de sessions que ens porti les dades de sessió mitjançant l'objecte `ModelAndView`. Aquest controlador és cridat des de GWT mitjançant l'objecte `RequestBuilder` i en quan obtenim la resposta, aquesta es tractada i enviada al servei GWT.

Vegem-ho en un exemple mitjançant aquest gràfic:

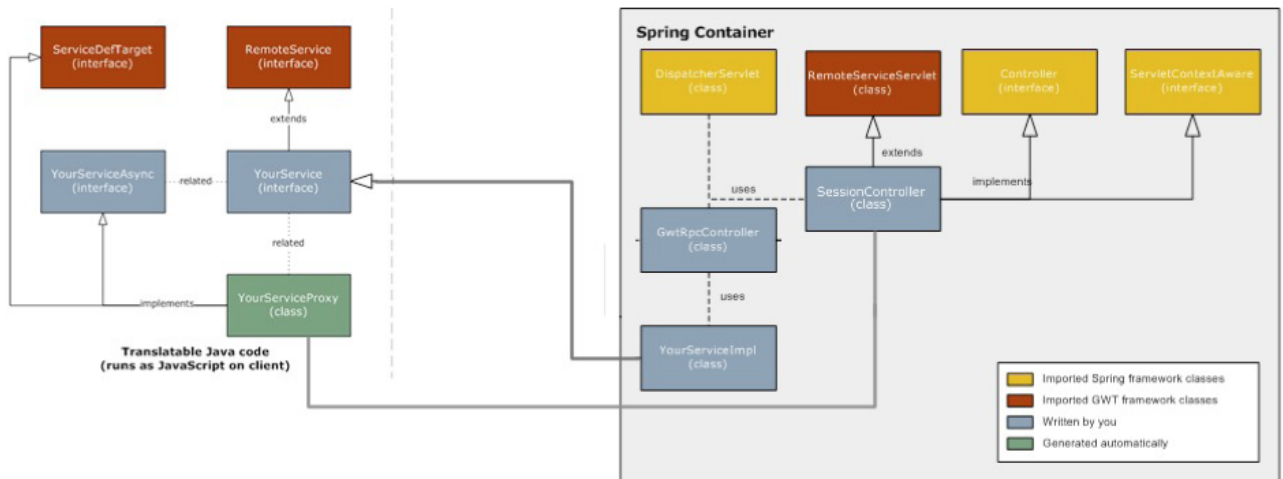


Figura 4.21: Comunicació indirecta

Cada vegada que tinguem que realitzar una crida a un servei que requereixi les dades de sessió, prèviament a la invocació del servei GWT farem una crida al controlador de sessió i un cop rebuda la resposta amb les dades de sessió actualitzades, realitzarem la invocació al servei RPC.

Les dades que rebem del controlador de sessió estan codificades en un document de text el qual ha de ser tractat per tal de poder treure les dades correctes.

Aquest és un procés molt lent, rudimentari i poc eficaç, per aquests motius en varem descartar la seva utilització.

Comunicació directa

La comunicació directa consisteix en enviar les dades directament des de GWTController al servei GWT, mitjançant una interfície que hereta de RemoteService i en modifica el seu contingut.

Per aconseguir aquest propòsit, s'han de fer unes petites modificacions tant a GWTController com al servei GWT, apart de crear una nova interfície SessionAware que serà l'encarregada de permetre la injecció de l'objecte HttpSession al servei GWT.

Anem a realitzar les modificació corresponents.

A GWTController afegim les següents línies dins del mètode processCall, entre les invocacions RPC.

```
public String processCall(String payload) throws SerializationException {
    try {
        RPCRequest rpcRequest = RPC.decodeRequest(payload, this.remoteServiceClass);
        /***** Línies a afegir *****/
        if (remoteService instanceof SessionAware) {
            ((SessionAware)remoteService).setSession(getRequest().getSession());
        }
        /***** */
        return RPC.invokeAndEncodeResponse(remoteService, rpcRequest.getMethod(), rpcRequest.getParam(), rpcRequest.getSerialization());
    }
    catch (IncompatibleRemoteServiceException ex)
    {
        getServletContext().log("An IncompatibleRemoteServiceException was thrown while processing this call.", ex);
        return RPC.encodeResponseForFailure(null, ex);
    }
}
```

Figura 4.22: Codi classe GWTController amb comunicació directa

A continuació crearem la interfície SessionAware on declararem un mètode per tal d'injectar l'objecte HttpSession al RemoteService.

```
public interface SessionAware{
    public void setSession(HttpSession session);
}
```

Figura 4.23: Codi classe SessionAware

Finalment implementarem el mètode setSession dins del codi d'implementació del servei GWT.

```
public class ServeiImpl extends RemoteServiceServlet implements Servei, SessionAware {
    private HttpSession Session;

    public void setSession(HttpSession session){
        Session = session;
    }

    // Altres metodes del servlet
}
```

Figura 4.24: Codi classe ServeiImpl

Amb aquest tipus de comunicació guanyem en rapidesa i en eficàcia, ja que la comunicació es molt més efectiva i fàcil.

Finalment podem observar en el gràfic següent l'estructura definitiva de la integració de GWT amb Spring.

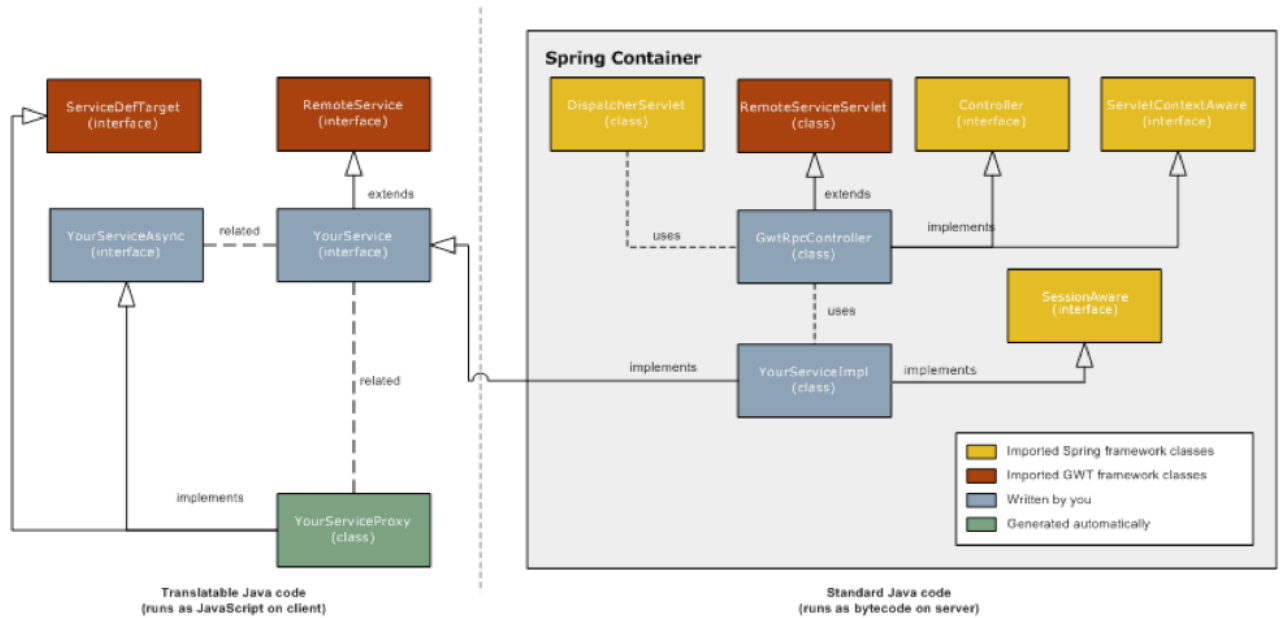


Figura 4.25: Comunicació directa

4.5 Particularitats de GWT en el projecte

4.5.1 Gestió de pantalles mitjançant GWT

Tenint en compte que InnovaCampus és una aplicació amb múltiples pantalles, en un principi es va decidir d'estructurar la interfície de l'aplicació en el format mòdul per pantalla. Tenint en compte que hi han diferents pantalles que comparteixen widgets, no era lògic replicar aquests widgets en cadascun dels mòduls ja que això implica un augment quasi exponencial del tamany de l'aplicació.

Per tal de solucionar aquesta problemàtica es va proposar unificar tots els components en un sol mòdul i crear un controlador de pantalla que s'encarregués de gestionar els components a activar.

Per tal de desenvolupar-ho es va decidir crear un identificador de pantalla dins de la sessió com un atribut nou, per tal de saber en tot moment en quina pantalla es troba l'usuari actiu. D'aquesta forma el gestor de pantalla pot saber quins són els widgets a activar hi ha desactivar, ja que s'identifica en temps real la pantalla activa..

L'esquema següent ens representa gràficament el funcionament del gestor de pantalla.

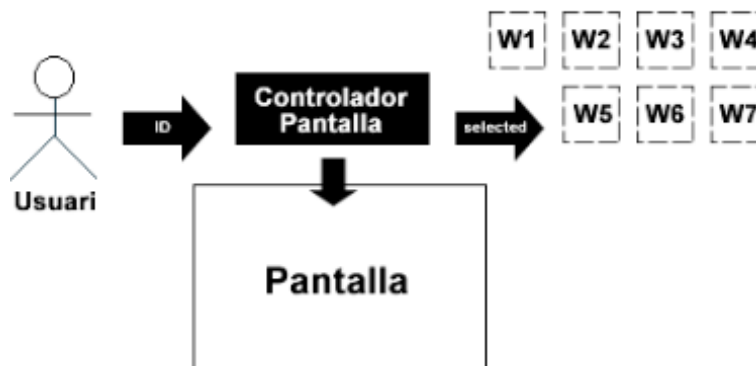


Figura 4.26: Gestió de pantalles i widgets

Funcionament de la gestió de pantalles:

El gestor de pantalla s'activa quan el navegador obre una nova pàgina web HTML. Un cop obert, el gestor realitza una petició al servidor per tal de comprovar l'estat actual de l'usuari.. Un cop identificada la pantalla on es troba l'usuari actiu, el controlador selecciona els widgets associats a la pantalla actual, aconseguint així que únicament es carreguin els widgets necessaris. Un cop identificats els widgets que cal carregar, és el mateix gestor qui els situa en el seu lloc corresponent dins la pàgina HTML.

L'identificador de pantalla:

Aquest element ens permet identificar en tot moment la situació real de l'usuari dins l'aplicació, o sigui, la direcció URL on està accedint l'usuari actiu, per tal d'informar al gestor si ens trobem a la pantalla d'inici, a la d'alumne, a la del professor o a la del administrador.

4.5.2 Aplicació del patró singleton

Durant la fase d'implementació de la lògica associada a la interfície gràfica hem fet ús del patró singleton. En enginyeria de programari, el patró singleton és un patró de disseny que es fa servir per a restringir la instanciació d'una classe a un objecte. Això és útil quan es necessita exactament un objecte per a coordinar accions a tot sistema, i això és el que ens passa en el disseny de les diferents classes que donen lloc a la interfície d'usuari d'InnovaCampus.

El patró singleton s'implementa creant una classe amb un mètode que crea una nova instància de la classe sempre i quan no n'existeixi cap. Si ja n'existeix una, simplement retorna una referència a aquell objecte. Per a estar segur de què l'objecte no es pot instanciar de cap altre manera, el constructor es fa o bé privat o bé protegit. Cal fer notar la distinció entre una simple instància estàtica d'una classe i un singleton: tot i que un singleton pot ser implementat com una instància estàtica, també pot ser construït mandròsament, per a no requerir memòria o recursos fins que es necessiti. Una altra diferència notable és que les classes membre estàtiques no poden implementar una interfície, a menys que la interfície sigui simplement un marcador. Així que si la classe ha de realitzar un contracte expressat per una interfície, s'ha de fer com un singleton.

4.5.3 Connexió amb la capa de negoci

Dins de la capa de presentació s'inclou tots els aspectes relacionats amb la interacció entre la capa de client i la capa de negoci, per tant, no únicament cal tenir en compte la interfície gràfica sinó també les crides que es fan al servidor i el seu processament, en conseqüència, GWT també ha d'implementar el que és coneix com a servlets RPC, que són els encarregats de rebre les peticions dels clients i de respondre al resultat de la petició. Donat que les crides RPC són asíncrones, també és necessari especificar el que s'anomena un Callback, on s'indiquen les accions a realitzar en cas que la petició hagi finalitzat correctament i també les accions en cas que la petició hagi finalitzat amb errors.

En el cas d'InnovaCampus, l'única finalitat d'aquests servlets serà la d'acabar processant una crida als serveis definits a la capa de negoci. Per tal de dur-ho a terme, cal implementar les classes que s'encarregaran de dur a terme aquestes crides, aquestes classes s'implementen com a part de la lògica executada pel servidor, no obstant, també requereixen especificar les corresponents classes que executarà el client.

Seguint el patró definit a la lògica de negoci, aquestes classes també s'han dividit en 4 subsistemes diferents:

- **AdministrationManagement:** Aquest subsistema s'encarrega de tasques de caire més administratiu, com per exemple la gestió d'usuaris, la creació d'assignatures o la gestió d'anuncis. S'encarrega de dur a terme totes les crides relacionades amb els serveis definits en el subsistema AdministrationManagement de la capa de negoci.
- **SubjectManagement:** La finalitat bàsica d'aquest subsistema és la gestió del contingut de les assignatures. S'encarrega d'afegir, modificar i esborrar els temes i objectius de les diferents assignatures. Una assignatura està dividida en temes i a la vegada els temes estan dividits en objectius. S'encarrega de dur a terme totes les crides relacionades amb els serveis definits en el subsistema SubjectManagement de la capa de negoci.
- **TestManagement:** La finalitat d'aquest subsistema és la de gestionar la creació de testos. És el responsable de la gestió de preguntes i respostes. Les preguntes formen part d'un objectiu en concret i cada pregunta té múltiples respostes. També s'encarrega d'associar recursos a preguntes

i de la generació de tests. S'encarrega de dur a terme totes les crides relacionades amb els serveis definits en el subsistema TestManagement de la capa de negoci.

- **StatisticsManagement:** Aquest subsistema és el que s'encarrega de gestionar la generació d'estadístiques i la consulta de tota la informació relacionada amb els resultats dels tests emmagatzemats. S'encarrega de dur a terme totes les crides relacionades amb els serveis definits en el subsistema StatisticsManagement de la capa de negoci.

Observem ara, un exemple d'implementació d'un dels serveis del subsistema SubjectManagement. Primerament s'implementa la lògica que hauria d'executar el servidor, que seria una crida al servei corresponent.

```
public String updateSubject(String subjectId, String name, String description)
{
    try{
        administrationManagement.updateSubject(subjectId, name, description);
    }catch (SubjectNotExistsException e){
        e.printStackTrace();
        return "ERROR_SUBJECT_NO_EXISTS";
    }
    return "OK_UPDATE_SUBJECT";
}
```

Figura 4.27: RemoteServiceImpl.java

El següent pas seria especificar les classes associades a la lògica del client. Per una banda es defineix la crida que s'ha d'executar al servidor i per altra s'especifica la corresponent crida asíncrona.

```
**
 * @gwt.typeArgs <java.lang.String>
 * @gwt.typeArgs subjectId <java.lang.String>
 * @gwt.typeArgs name <java.lang.String>
 * @gwt.typeArgs description <java.lang.String>
 */
String updateSubject(String subjectId, String name, String description);
```

Figura 4.28: RemoteServices.java

```
void updateSubject(String subjectId, String name, String description, AsyncCallback<String> callback);
```

Figura 4.29: RemoteServiceAsync.java

4.5.4 Gestió de l'idioma mitjançant GWT + i18n

Actualment InnovaCampus està disponible en tres idiomes diferents i totalment funcionals: català, castellà i anglès, no obstant, cal tenir en compte que en un futur pot esdevenir necessari portar l'aplicació a altre idiomes, per tant, és imprescindible que l'aplicació sigui flexible a l'idioma, és aquí on entra en joc el procés d'internacionalització. La internacionalització és el procés de dissenyar programari de tal manera que pugui adaptar-se a diferents idiomes i regions sense la necessitat realitzar canvis en el codi del programa.

És una pràctica comuna en l'anglès (sobretot en l'àmbit de la computació), abreviar internationalization com "i18n". Això es degut al fet que entre la primera i l'última n de la citada paraula hi ha 18 lletres.

Per solucionar el problema de la internacionalització hem decidit fer servir missatges externs. Com que es molt fàcil que aparegui el mateix text en varies pàgines web, existeix la possibilitat de que cada ocurrència sigui inconsistent amb la resta, és a dir, que s'usin noms diferents en diferents llocs per referir-se al mateix concepte. També podria passar que es decidís canviar el text, llavors el canvi afectaria a cada pàgina en la que aparegués el text en concret.

Per tractar aquest problemes, GWT ofereix una sèrie de recursos i eines que descriurem a continuació.

Modul i18N

Google Web Toolkit conté un conjunt de classes dedicades al procés d'internacionalització que nosaltres haurem d'utilitzar per tal d'aplicar-ho a la nostra aplicació.

Aquestes classes es troben ubicades dins del mòdul 'com.google.gwt.i18n.I18N'. La següent instrucció ens permet afegir el mòdul dins del mòdul implementat per a InnovaCampus.

```
<inherits name="com.google.gwt.i18n.I18N" />
```

Figura 4.30: Modul i18N

Classe ICLangConst

La classe ICLangConst, abreviatura de InnovaCampus Language Constants, contindrà una conjunt de funcions sense paràmetres d'entrada i amb un String de retorn per a cadascuna de les funcions. Aquestes funcions fan referència directa a una constant dels fitxers de propietats de llenguatge ICLangConst.properties, on es troba ubicat el text real que apareixerà a la interfície.

```
public interface ICLangConst extends Constants{

    /* AUTHFORM */
    String TitleIniAuth();
    String LoginText();
    String PasswText();
    String ButtonAuthText();
    String NoUserAccesText();|
}
```

Figura 4.31: ICLangConst.java

Classe ICLangMessg

La classe ICLangMessg abreviatura de InnovaCampus Language Messages, contindrà un conjunt de funcions amb paràmetres d'entrada i amb un String de retorn per a cadascuna d'elles. Aquestes funcions fan referència directa a un missatge dels fitxers de propietats de llenguatge ICLangMessg.properties on s'ubica el missatge real que apareixerà a la interfície.

```
public interface ICLangMessg extends Messages{

    /* AUTHFORM */
    String Error3(String UserName);
}
```

Figura 4.32: ICLangMessg.java

Fitxers de propietats de llenguatge

Són els fitxer que contindran el text real de les constats i dels missatges associats a un idioma determinat.

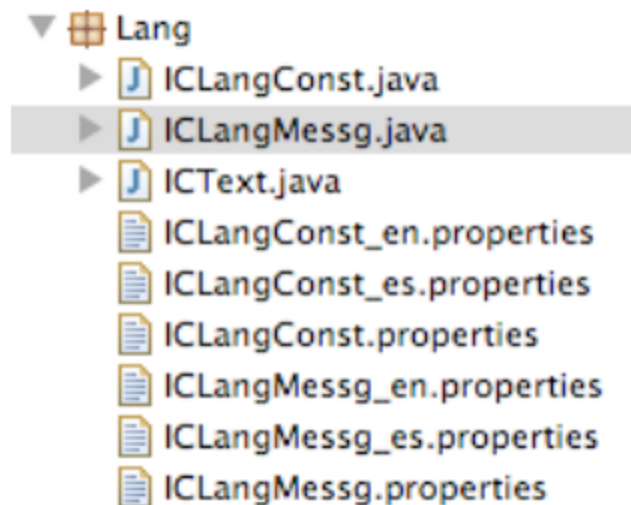


Figura 4.33: Package Lang d'InnovaCampus

Per tal d'indicar a quin idioma pertany cada fitxer es nombraran mitjançant les sigles d'abreviatura d'idiomes internacional, tot plegat permetrà que l'aplicació els pugui identificar.

Exemple de fitxers de propietats:

```
ICLangConst_es.properties

$PP_OFF
# Spanish Language
#-----

#--- AUTHFORM
TitleIniAuth      = Iniciar sesión en
LoginText          = DNI o NIE:
PasswText          = Contraseña:
ButtonAuthText     = Inicio sesión
NoUserAccesText    = No puedo acceder a mi cuenta!

ICLangMessg_en.properties

$PP_OFF
# English Language
#-----

#--- AUTHFORM
Error3             = User '{0}' incorrect password
```

Figura 4.34: Fitxers de propietats

El símbol '{0}' indica que el contingut del paràmetre numero 0 de la funció Error3 anirà situat allí.

La variable “locale”

Aquesta és la variable que ens permet escollir el llenguatge amb el que volem visualitzar l'aplicació. Aquesta variable s'envia a la pàgina en format GET (dins de la URL) per tal que el citat mòdul i18N de GWT s'encarregui de determinar el corresponent fitxer de propietats de llenguatge.

Resumint, si la nostra aplicació rep una petició on la variable locale té assignat el valor 'en' (login.htm?locale=en) el sistema procedirà a modificar el text a l'anglès a partir del del fitxer de propietats de llenguatge '..._en.properties'.

A més a més, per tal de mantenir l'idioma de navegació durant la sessió s'ha incorporat a la entitat usuari una nova variable anomenada 'Lang' que s'encarrega d'enregistrar l'idioma escollit.

Podeu trobar més informació relacionada amb la internacionalització a [33].

4.5.5 Extensions de GWT utilitzades

La capa de presentació d'InnovaCampus ha estat totalment desenvolupada amb GWT per tal d'aconseguir una experiència més interactiva i atractiva per a l'usuari final.

També s'ha utilitzat una extensió de GWT que ens ha permès donar-li un aspecte més professional a les interfícies, facilitant-nos així la feina de disseny i maquetació ja que aquesta extensió aporta widgets ja creats per tal de que puguin ser integrats a les noves aplicacions.

4.5.6 El paper del Tiles

Apache Tiles és un marc de plantilles dissenyat per simplificar el desenvolupament d'interfícies web d'usuari.

Tiles permet als autors definir fragments de pàgina actualitzables en temps d'execució cosa que afavoreix la reutilització i el desenvolupament d'una aparença consistent .

El mòdul d'estadístiques d'InnovaCampus fa ús de plantilles creades mitjançant aquest marc, tot seguit en mostrem un exemple:

```
<html> [...]
  <body>
    <div id="web-frame">
      <!--capçalera--> <tiles:insertAttribute name="header"/>
      <!-- Menu principal --> <tiles:insertAttribute name="main-menu"/>
      <!--Contingut--> <div id="content-frame">[...] </div>
      <!--fi content frame--> <hr /> <!--footer--> <div id="footer">
        <tiles:insertAttribute name="footer"/> </div>
      <!--fi footer--> </div>
    </body>
  </html>
```

Figura 4.35: Exemple HTM amb etiquetes Tiles

S'utilitza l'etiqueta JSP Tiles `<tiles:insert>` per incloure contingut a la plantilla. Els detalls d'on s'origina el contingut inclòs s'especifiquen al fitxer de configuració de Tiles, fitxer XML que descriu com omplir la plantilla. El següent fragment mostra una part de la plantilla principal (anomenada template), omplint cadascun dels components:

```

<definition name="template" template="/WEB-INF/jsp/viewTemplate.jsp">
    <put-attribute name="title" value="Innova Campus"/>
    <put-attribute name="icon" value="assets/img/logoNav.ico"/>
    <put-attribute name="page-type" value="start"/>
    <put-attribute name="gwt" value="gwt/gwt.nocache.js"/>
    <put-attribute name="gwttext1" value="assets/gwttext/adapters/ext/ext-base.js"/>
    <put-attribute name="gwttext2" value="assets/gwttext/ext-all.js"/>
    <put-attribute name="css1" value="assets/css/stylesHTML.css"/>
    <put-attribute name="css2" value="assets/css/stylesGWT.css"/>
    <put-attribute name="css3" value="assets/gwttext/resources/css/ext-all.css"/>
    <put-attribute name="css4" value="assets/css/styleTreePanel.css"/>
    <put-attribute name="css5" value="assets/css/datePicker.css"/>
    <put-attribute name="header" value="/WEB-INF/jsp/header.jsp"/>
    <put-attribute name="content" value="/WEB-INF/jsp/content.jsp"/>
    <put-attribute name="footer" value="/WEB-INF/jsp/footer.jsp"/>
    <put-attribute name="InnovaCampusLogo" value="assets/img/logo.png"/>
</definition>

```

Figura 4.36: Definició d'un templates

Aquí els components header, content i footer prendran la ruta a fitxers JSP que defineixen com han de ser l'encapçalament, el contingut i el peu de pàgina. Quan Tiles construeixi la pàgina, substituirà les etiquetes `<tiles:insert>` anomenades header, content i footer amb la informació de sortida de header.jsp, content.jsp i footer.jsp, respectivament.

A més a més, també s'indiquen els fitxers JavaScript (.js) que genera el compilador de Google Web Toolkit. Aquests fitxers són els identificats anteriorment com a "ext-base.js", "ext-all.js" i "gwt.nocache.js". Observem que els hem identificat com la resta de fitxers:

```

<put-attribute name="gwt" value="gwt/gwt.nocache.js"/>
<put-attribute name="gwttext1" value="assets/gwttext/adapters/ext/ext-base.js"/>
<put-attribute name="gwttext2" value="assets/gwttext/ext-all.js"/>

```

Figura 4.37: Definició de subtemplates

A partir de la plantilla principal "template" s'aniran originant diferents pàgines JSP en funció de les rutes que es defineixen als components. En els següents exemples, s'especifiquen les rutes que s'associaran a les pàgines d'inici de l'administrador, d'un estudiant i d'un professor respectivament:

```

<!-- admin pages -->

<definition name="adminHome" extends="template">
    <put-attribute name="page-type" value="admin"/>
    <put-attribute name="title" value="Innova Campus | Administration Home"/>
    <put-attribute name="content" value="/WEB-INF/jsp/admin/content.jsp"/>
</definition>

<!-- student pages -->

<definition name="studentHome" extends="template">
    <put-attribute name="page-type" value="student"/>
    <put-attribute name="title" value="Innova Campus | Student Home"/>
    <put-attribute name="content" value="/WEB-INF/jsp/student/content.jsp"/>
</definition>

<!-- teacher pages -->

<definition name="teacherHome" extends="template">
    <put-attribute name="page-type" value="teacher"/>
    <put-attribute name="title" value="Innova Campus | Teacher Home"/>
    <put-attribute name="content" value="/WEB-INF/jsp/teacher/content.jsp"/>
</definition>

```

Figura 4.38: Definició de subtemplates

Observem que es tracta de definicions marcades com a extensió de la plantilla principal (atribut `extends="template"`), cosa que garanteix que la pàgina heretarà totes les definicions dels seus components i permetrà redifinir els components necessaris en funció de la pàgina creada. En aquest cas observem que es redefeixen els continguts associats al tipus de pàgina, títol i contingut per tal de personalitzar les pàgines dels diferents perfils d'usuari.

Per integrar Tiles a Spring MVC se li ha de dir a Spring que obri el fitxer(s) de configuració de Tiles. Spring disposa d'un bean que gestiona aquets fitxers i els prepara per mostrar vistes Tiles. Aquest bean s'anomena `TilesConfigurer` i conté la propietat `definitions` que defineix la ruta al fitxer de configuració de Tiles, tal com es mostra a continuació:

```

<bean id="tilesConfigurer" class="org.springframework.web.servlet.view.tiles2.TilesConfigurer">
    <property name="definitions" value="/WEB-INF/InnovaCampus-tiles.xml"/>
</bean>

```

Figura 4.39: Configuració dels Tiles

Podeu trobar més informació del marc Apache Tiles a [Annex A] i [31]

Capítol 5

Disseny: Altres aspectes rellevants

5.1 Seguretat

Al llarg dels capítols 3 i 4 hem fet una aproximació detallada a l'arquitectura de l'aplicació, no obstant, no hem d'oblidar-nos d'altres aspectes fonamentals en el món de les aplicacions Web com per exemple la seguretat.

La seguretat, en el món de la informàtica, així com en altres àrees, es basa en la protecció d'actius. Aquests actius poden ser elements tan tangibles com un servidor o una base de dades. Generalment podem avaluar la seguretat d'un actiu en base a tres aspectes principals: integritat, disponibilitat i confidencialitat.

Aquests tres aspectes, al seu torn, depenen de tres elements principals que engloben pràcticament tots els diferents controls que es poden establir en un sistema informàtic:

- Autenticació: els clients de les nostres aplicacions o serveis han de ser identificats de forma única, siguin usuaris finals, altres serveis o ordinadors externs.
- Autorització: no només és necessari saber qui accedeix als nostres actius, també cal establir que és el que poden fer amb ells. Un nivell d'autorització donat determina quin tipus d'operacions o transaccions pot realitzar un client sobre un determinat recurs.
- Registre i Auditoria: després d'haver realitzat una acció, és important que aquesta sigui registrada adequadament, en particular és essencial si volem evitar el repudi de transaccions efectuada per un client.

Tots aquests conceptes són especialment vàlids a l'entorn d'Internet, i particularment importants donat el creixement exponencial dels serveis i aplicacions accessibles a través de la Web, així doncs, en aquest capítol detallarem els components de seguretats emprats en el disseny d'InnovaCampus.

5.1.1 Seguretat a InnovaCampus: SpringSecurity

Per tal d'aportar un nivell alt de seguretat al projecte InnovaCampus, des de l'inici es va apostar per la utilització de Spring Security, un dels frameworks de seguretat web més eficients i segurs que existeixen avui en dia.

Spring Security (basat en el projecte Acegi Security) proporciona a les aplicacions webs un sistema estable d'autorització i autenticació d'usuaris. Una de les principals característiques d'aquest framework és que no és invasiu, és a dir, no és obligatòria la modificació del codi en la seva utilització, per tant permet una integració fàcil i ràpida.

Mitjançant aquest framework, aconseguirem dotar a l'aplicació InnovaCampus de seguretat en els processos d'autenticació i autorització d'usuaris.

Per més informació sobre Spring Security consultar [29] i [37].

5.1.1.1 Control d'autoritzacions

L'autorització és refereix al procés de decidir si un usuari està autoritzat a fer una acció o no. Per tant, a InnovaCampus el control d'autoritzacions, s'encarregarà de controlar l'accés dels usuaris en certs llocs segons el seu rol.

Per tal de dur-ho a terme s'han introduït una sèrie de filtres dins la configuració de seguretat de l'aplicació.

```
<intercept-url pattern="/login.htm" filters="none" />
<intercept-url pattern="/admin/**" access="ROLE_A" />
```

Figura 5.1: Interceptors SpringSecurity

Amb això aconseguirem que a la pàgina login hi pugui accedir qualsevol usuari. En canvi dins de la carpeta admin i per tant a qualsevol de les pàgines que contingui aquesta carpeta, únicament podran accedir-hi aquells usuaris que tinguin el rol d'administrador.

5.1.1.2 Procés d'autenticació

Tenint en compte que l'encarregat de realitzar el procés d'autenticació d'usuaris és Spring Security i que això es realitza mitjançant una crida estàndard (no RPC) al servlet `'j_spring_security_check'`, ens hem vist obligats a redefinir el flux implementat per tal de fer ús del dinamisme que ens proporcionen les crides asíncrones. Per tant, el que fa el servlet RPC (que fa ús dels serveis d'autenticació) és fer una nova petició al servidor per tal d'autenticar l'usuari fent ús del propi sistema d'autenticació de Spring Security.

Spring Security disposa d'un gran nombre de constats on es guarda part de la informació que es genera en el procés d'autenticació. Aquestes constants les podeu trobar especificades a la següent adreça: Vegeu [21].

Per més informació consultar [Annex A].

5.1.2 Altres aspectes

5.1.2.1 La codificació de la clau d'accés

Un altre aspecte important és el de la codificació de la clau d'accés. Aquesta clau d'accés seria convenient que no estés guardada en pla a la base de dades. S'ha de codificar. Anem a veure el procediment que hem seguit per tal d'aconseguir-ho.

No ens ha fet falta encriptar les claus. Simplement, generar un hash MD5 o SHA1 (millor SHA1, que és més segur). Aquestes funcions hash tenen la particularitat que són d'una sola via: a partir d'un text pots obtenir el codi hash, però no a la inversa (no és possible obtenir el text a partir del codi hash). A més a més, és virtualment impossible aconseguir dos textos que generin el mateix codi hash. Una altra peculiaritat d'aquest sistema és que els codis hash generats no són text pla, sinó un conjunt arbitrari de bits, això sí, de longitud fixa.

Per guardar-los a la base de dades, en un camp de tipus text, convé donar a aquests codis un petit tractament, que consisteix en transformar-los a BASE64 (que sí són caràcters ASCII).

El sistema és el següent:

- A la base de dades no guardem les claus, sinó els seus codis MD5 o SHA1 traduïts a BASE64.
- Quan es vol comprovar una parella nom d'usuari – clau, s'obté el codi MD5 o SHA1 de la clau, es passa a BASE64 i el compara amb el que té emmagatzemat per a aquell nom d'usuari.
- Si coincideix: la clau és correcta.
- Si no coincideix, doncs és incorrecta.

Podeu trobar més informació sobre les codificacions MD5 i SHA1 a [24] i [25] respectivament.

5.1.2.2 Control del registre d'estudiant

Tots els estudiants que es registrin al sistema han d'haver estat matriculats prèviament en alguna de les assignatures. Si no és així no es podrà dur a terme el registre al sistema.

Per tal de registrar-se l'estudiant haurà d'omplir un formulari on se li demana el DNI, el correu electrònic i la clau d'accés. Si el DNI que introdueix l'usuari no ha estat donat d'alta prèviament per un professor al fer la matrícula, l'alta no es durà a terme, en cas contrari l'alta es produirà amb normalitat i es procedirà a l'enviament d'un correu electrònic a l'usuari amb les dades corresponents.

Aquest procés té un petit problema de seguretat. Si algun alumne coneix el DNI d'algun altre pot registrar-se al sistema fent-se passar per ell. La nostra aplicació inclou una opció de configuració que força als usuaris a introduir una adreça de correu electrònic que respongui a un domini determinat. D'aquesta manera totes aquelles institucions que utilitzin el software i ofereixin comptes de correu als seus alumnes podran controlar a tots els que intentin registrar-se amb les dades d'un altre.

Per tal de controlar aquesta restricció del domini de correu ens hem decantat per definir paràmetres dins del fitxer web.xml. L'encarregat de definir-los serà el desplegador de l'aplicació. Observem a continuació un exemple del codi que s'hauria d'afegir al fitxer web.xml:

```
<context-param>
    <param-name> emaildomains </param-name>
    <param-value> alumnes.eps.udl.es </param-value>
</context-param>
```

Figura 5.2: Context-param per restringir domini de correu

Podrem recuperar el valor del paràmetre “emaildomains” mitjançant la setència `getServletContext().getInitParameter("emaildomains")`.

D'aquesta manera podrem controlar la restricció del domini de correu electrònic dins del Filtre que s'encarrega de gestionar el registre d'un alumne. Si el valor que conté aquest paràmetre és ANY indica que s'accepta qualsevol domini.

Capítol 6

Disseny de la interfície gràfica

6.1 Anàlisi jeràrquic de tasques (HTA)

L'anàlisi jeràrquic de tasques (HTA Hierarchical Task Analysis) desenvolupat per Annett i Duncan, és la tècnica d'anàlisi de tasques més conegut i antic, que continua essent vigent encara que hagin aparegut noves tecnologies.

L'objectiu del HTA és realitzar una descripció de tasques en termes d'operacions i plans. Les operacions (descomposició en sub-tasques) són activitats que realitzen les persones per aconseguir un objectiu. Els plans són una descripció de les condicions que s'han de donar quan es realitza cadascuna de les activitats.

Les operacions es poden descompondre de forma jeràrquica i s'assigna un pla a cadascuna de les tasques que apareixen. Es defineix un objectiu com un estat determinat del sistema al qual vol arribar un usuari/a.

La descripció de la informació es realitza en forma de taula o en forma de diagrama d'arbre que descriu les relacions entre tasques i subtasques.

A continuació podem veure 4 exemples d'alguns dels HTA que hem realitzat.

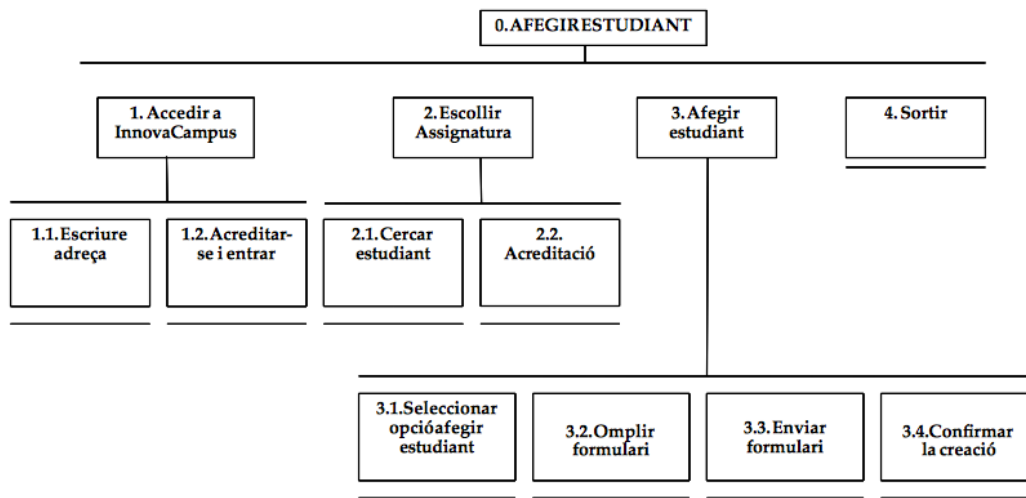


Figura 6.1: HTA Afegir estudiant

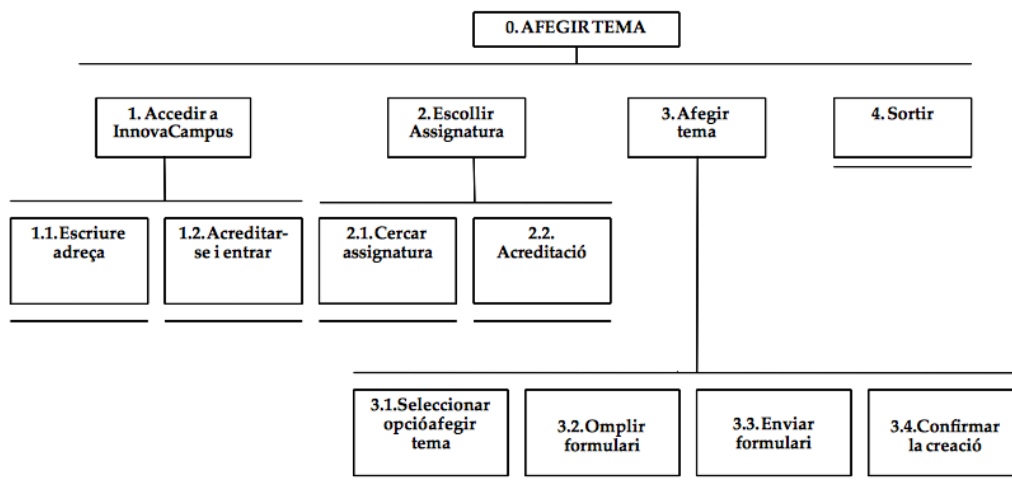


Figura 6.2: HTA Afegir tema

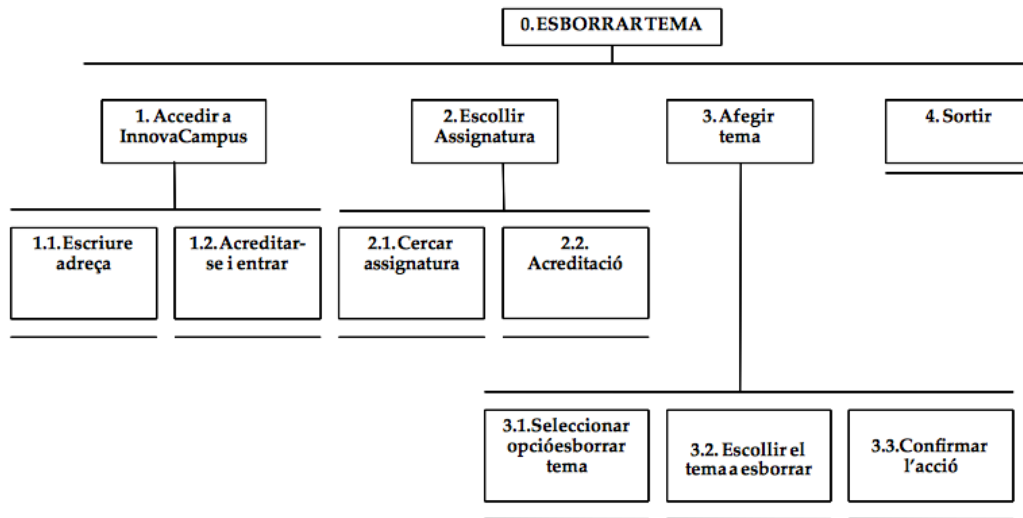


Figura 6.3: HTA Esborrar tema

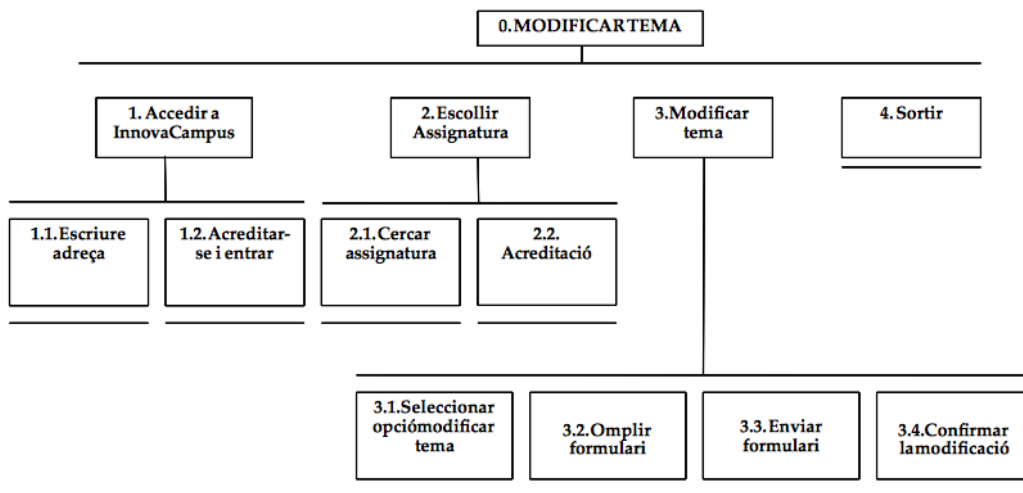


Figura 6.4: HTA Modificar tema

6.2 Arquitectura de la informació

La informació és una font de coneixement. Però, si no està organitzada, processada i disponible per a les persones en un format que permeti prendre decisions, més que un benefici és una nosa.

L'arquitectura de la informació es refereix a l'estructura de la organització de l'aplicatiu, especialment en com les diferents pantalles de l'aplicació es troben relacionades entre elles. Implica opcions com la planificació, l'anàlisi dels continguts, organització de les pantalles, proporcionar indicadors d'ajuda i orientació, etiquetatge i disseny de la navegació.

La mateixa informació pot tenir diferents estructures raonables depenent de com la gent pensa, parla o utilitza l'aplicació.

Utilitzarem la informació aportada per l'anàlisi de requisits i l'anàlisi de tasques. Això ens permetrà disposar de peces de contingut potencial, etiquetes i esquemes d'organització.

6.2.1 Identificació d'objectius

En l'elaboració dels continguts de qualsevol aplicació el primer que hem de fer és procedir amb la informació dels objectes o unitats d'informació que contindrà la nostra aplicació.

Hem de tenir en compte que ja partim d'un disseny anterior i per tant ja disposem d'una estructura d'informació preestablerta. Tot i així l'analitzarem i en realitzarem modificacions si ho creiem oportú.

Per tal de definir els nostres continguts i estructurar la informació es varen realitzar diferents reunions amb l'equip de desenvolupament i es varen realitzar anàlisis de l'aplicatiu InnovaCampus 1.0.

- Rol de Professor
 - Configuració de les dades de perfil d'usuari
 - Gestió d'assignatures
 - Llista d'assignatures disponibles
 - Anuncis
 - Gestionar professors de les assignatures
 - Gestionar els temes de les assignatures
 - Gestionar els objectius de les assignatures
 - Gestionar les preguntes de les assignatures
 - Gestionar les respostes de les assignatures
 - Gestionar els alumnes de les assignatures
 - Gestionar els anuncis privats de cada assignatura
 - Visualitzar les estadístiques de les assignatures

- Rol d'Alumne
 - Configuració de les dades del perfil d'usuari
 - Fer un test
 - Llista d'assignatures disponibles Anuncis
- Rol d'Administrador
 - Configuració de les dades del perfil d'usuari
 - Gestió de professors
 - Gestió d'assignatures
 - Gestió d'anuncis genèrics

6.2.2 Ordenació de Targetes (Card-Sorting)

Un cop identificats els objectes ens trobem davant el repte d'organitzar-los de manera que sigui útil i comprensible per als usuaris de la nova aplicació.

La tècnica coneguda com ordenació de targetes o Card Sorting, resulta altament útil per conèixer com els usuaris visualitzen la organització de la informació. La idea serà utilitzar les aportacions dels usuaris per decidir com haurem d'estructurar la informació.

El Card Sorting és una tècnica simple, barata, ràpida i que involucra els usuaris. Per tal d'implementar una ordenació de targetes hem de seguir els següents passos:

- Determinar la llista de tòpics.
- Crear les targetes. Cada tòpic haurà d'escriure's en una targeta. Per a simplificar el procés utilitzarem un programa que ens ajudarà. Es tracta del programa anomenat CardZort de Jorge A. Toro.
- Seleccionar els participants. Hem escollit a 4 participants que ens faran les ordenacions de les targetes que els facilitarem. Cadascú farà l'ordenació en un ordinador. Els participants són la Marta, el Josep Maria, l'Antoni i el Didac.
- Procedir amb la sessió d'ordenació. Cada usuari fa la seva ordenació en un ordinador diferent, organitza les targetes i etiqueta els grups segons els seus propis criteris personals.
- Les ordenacions i conclusions que es treuen a continuació són referents al menú superior de la pantalla general o d'escriptori.

Els resultat obtingut del cardzorting realitzat per 3 usuaris ha estat el següent:

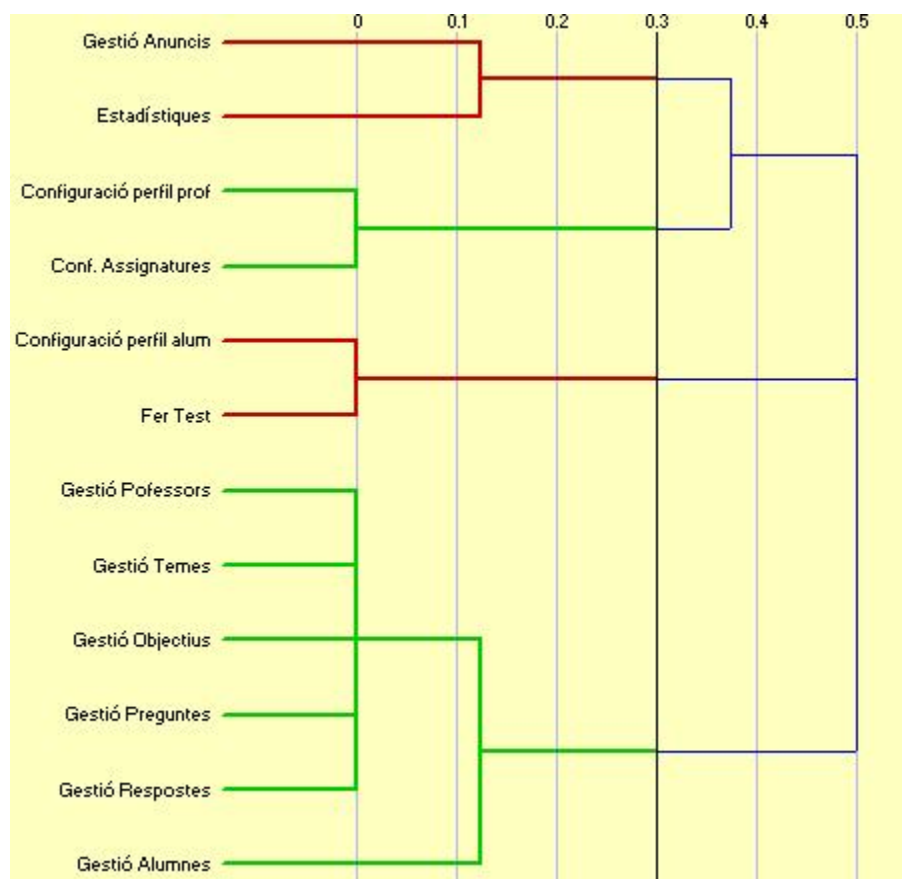


Figura 6.5: Cluster cardsorting

Si analitzam els resultats podem observar com l'ordenació mitjana que han realitzat els usuaris a 0.3 ens genera una organització molt semblant a la que teníem a InnovaCampus 1.0, per tant després d'analitzar-ho i valorar-ho més profundament hem obtingut la següent ordenació de menús per a la nova aplicació.

InnovaCampus 1.0	InnovaCampus 2.0
<ul style="list-style-type: none"> ☒ Professor <ul style="list-style-type: none"> ◦ Configuració <ul style="list-style-type: none"> ▪ Canvi password ▪ Canvi dades ◦ Assignatura <ul style="list-style-type: none"> ▪ Afegir ▪ Esborrar ▪ Editar ▪ Reiniciar ◦ Tancar sessió 	<ul style="list-style-type: none"> ☒ Professor <ul style="list-style-type: none"> ◦ Configuració <ul style="list-style-type: none"> ▪ Perfil ▪ Modificar Contrasenya ◦ Assignatura <ul style="list-style-type: none"> ▪ Afegir ▪ Editar
<ul style="list-style-type: none"> ☒ Professor - Assignatura <ul style="list-style-type: none"> ◦ Professors <ul style="list-style-type: none"> ▪ Afegir ▪ Consultar ▪ Esborrar ◦ Temes <ul style="list-style-type: none"> ▪ Afegir ▪ Editar ▪ Consultar ▪ Esborrar ◦ Objectius <ul style="list-style-type: none"> ▪ Afegir ▪ Editar ▪ Consultar ▪ Esborrar ◦ Preguntes / Recursos <ul style="list-style-type: none"> ▪ Afegir ▪ Editar ▪ Consultar ▪ Esborrar ◦ Alumnes <ul style="list-style-type: none"> ▪ Afegir des de teclat ▪ Afegir des de fitxer ▪ Consultar ▪ Esborrar ◦ Anuncis <ul style="list-style-type: none"> ▪ Afegir ▪ Consultar ▪ Esborrar ◦ Estadístiques <ul style="list-style-type: none"> ▪ Objectiu estudiant ▪ Detallades 	<ul style="list-style-type: none"> ☒ Professor - Assignatura <ul style="list-style-type: none"> ◦ Professors <ul style="list-style-type: none"> ▪ Afegir ▪ Assignar ▪ Consultar ◦ Alumnes <ul style="list-style-type: none"> ▪ Afegir des de teclat ▪ Assignar ▪ Consultar ▪ Importar ▪ Exportar ◦ Temes <ul style="list-style-type: none"> ▪ Afegir ▪ Consultar ◦ Objectius <ul style="list-style-type: none"> ▪ Afegir ▪ Consultar ◦ Preguntes <ul style="list-style-type: none"> ▪ Afegir ▪ Consultar ▪ Importar ▪ Exportar ◦ Respostes <ul style="list-style-type: none"> ▪ Afegir una resposta ◦ Anuncis <ul style="list-style-type: none"> ▪ Afegir ▪ Borra ◦ Estadístiques

Figura 6.6: Menús antics i Menús actuals

<ul style="list-style-type: none"> ☒ Alumne <ul style="list-style-type: none"> ◦ Configuració <ul style="list-style-type: none"> ▪ Canviar password ▪ Canviar dades ◦ Tancar sessió 	<ul style="list-style-type: none"> ☒ Alumne <ul style="list-style-type: none"> ◦ Configuració <ul style="list-style-type: none"> ▪ Perfil ▪ Modificar Contrasenya
<ul style="list-style-type: none"> ☒ Alumne-Assignatura <ul style="list-style-type: none"> ◦ Test <ul style="list-style-type: none"> ▪ Resoldre ◦ Anuncis <ul style="list-style-type: none"> ▪ Veure anuncis ◦ Tancar sessió 	<ul style="list-style-type: none"> ☒ Alumne-Assignatura <ul style="list-style-type: none"> ◦ Test <ul style="list-style-type: none"> ▪ Fer un test
<ul style="list-style-type: none"> ☒ Administrador <ul style="list-style-type: none"> ◦ Configuració <ul style="list-style-type: none"> ▪ Canviar Password ◦ Professor <ul style="list-style-type: none"> ▪ Afegir ▪ Esborrar ◦ Anunci <ul style="list-style-type: none"> ▪ Afegir ▪ Esborrar 	<ul style="list-style-type: none"> ☒ Administrador <ul style="list-style-type: none"> ◦ Configuració <ul style="list-style-type: none"> ▪ Perfil ▪ Modificar Contrasenya ◦ Professors <ul style="list-style-type: none"> ▪ Afegir ▪ Editar ▪ Assignar ▪ Importar ▪ Exportar ◦ Assignatures <ul style="list-style-type: none"> ▪ Editar ▪ Esborrar

Figura 6.7: Menús antics i Menús actuals

6.2.3 Estructura i organització de continguts

L'estudi dels models de navegació permet determinar i entendre com naveguem els usuaris per definir com volem que naveguin pel nostre lloc.

Hem de tenir clar que l'usuari utilitza com a referència una pàgina d'inici i un menú constant, llegit en diagonal (és a dir no llegeix detalladament, sinó que fa una mena d'escaneig ràpid), per tant estructurarem el contingut de les noves interfícies basant-nos en una estructura jeràrquica on l'usuari trobarà una pantalla inicial i a partir d'aquí podrà accedir a la resta de pantalles de l'aplicació.

La pantalla inicial o de benvinguda a l'aplicació ens permetrà accedir les diferents interfícies de nivell dos, segons el rol d'usuari que sigui.

- Interfície d'administració del sistema
- Interfície de professorat
- Interfície d'alumne

Cadascuna d'aquestes interfícies ens permetran accedir a les interfícies de nivell tres corresponents a les funcionalitats que pot realitzar cada usuari.

- Interfície d'administració del sistema
 - Interfícies de configuració de perfil
 - Interfícies de gestió de professorat
 - Interfícies de gestió de assignatures
 - Interfícies de gestió d'anuncis
- Interfície de professorat
 - Interfícies de configuració de perfil
 - Interfícies de gestió de professorat
 - Interfícies de gestió de assignatures
 - Interfícies de gestió de temes
 - Interfícies de gestió de objectius
 - Interfícies de gestió de alumnes
 - Interfícies de gestió de preguntes
 - Interfícies de gestió de respostes
 - Interfícies de gestió de estadístiques
 - Interfícies de gestió d'anuncis
- Interfície de l'alumnat
 - Interfícies de configuració de perfil
 - Interfícies de test

6.2.4 Navegació i Pantalles

Un cop definida la topologia del contingut a partir de l'estructura aportada per l'ordenació de targetes, completarem aquesta feina afegint dreceres aportades per l'anàlisi de tasques, desenvoluparem les diferents barres de navegació i les senyals d'orientació. Els principals objectius de la navegació són:

- Permetre als usuaris trobar la trajectòria més fàcil per arribar al seu objectiu.
- Assegurar que els usuaris saben sempre on són.
- Assegurar que els usuaris es mouen ràpidament i de manera lògica i de forma atractiva per l'aplicació.
- Donar als usuaris un context del que estan llegint.
- Destacar als usuaris parts de l'aplicatiu que ens interessa promocionar.

Pantalla Inicial

La pantalla inicial està dividida en dues parts, la part superior amb el selector d'idioma i la part inferior on hi podem localitzar el taulell d'anuncis i els formularis d'autenticació i registre d'usuaris nous.



Figura 6.8: Part superior

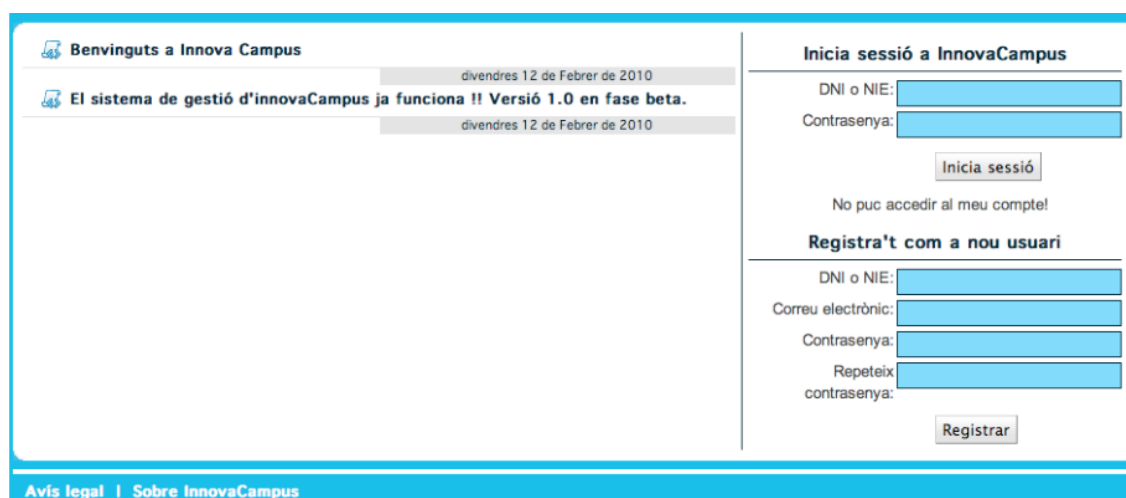


Figura 6.9: Part inferior pantalla inicial

Pantalla d'usuari

La pantalla d'usuari ens permet estructurar de forma automàtica mitjançant GWT les 3 pantalles referents als tres rols d'usuari que existeixen a l'aplicació (Alumne, Professor, Administrador).

Pantalla d'usuari alumne

La pantalla de l'usuari alumne s'organitza en quatre part, el menú d'opcions, el menú d'assignatures on està matriculat l'alumne, un taulell d'anuncis i la part central on es presenten les diferents opcions seleccionades des del menú en forma d'arbre.

Quan l'usuari selecciona una assignatura aquesta queda marcada en color gris fosc, per tal de deixar entendre que aquesta és l'assignatura que està activa, si volem canviar sols cal desactivar-la tornant a prémer d'amunt i automàticament es desmarcarà.

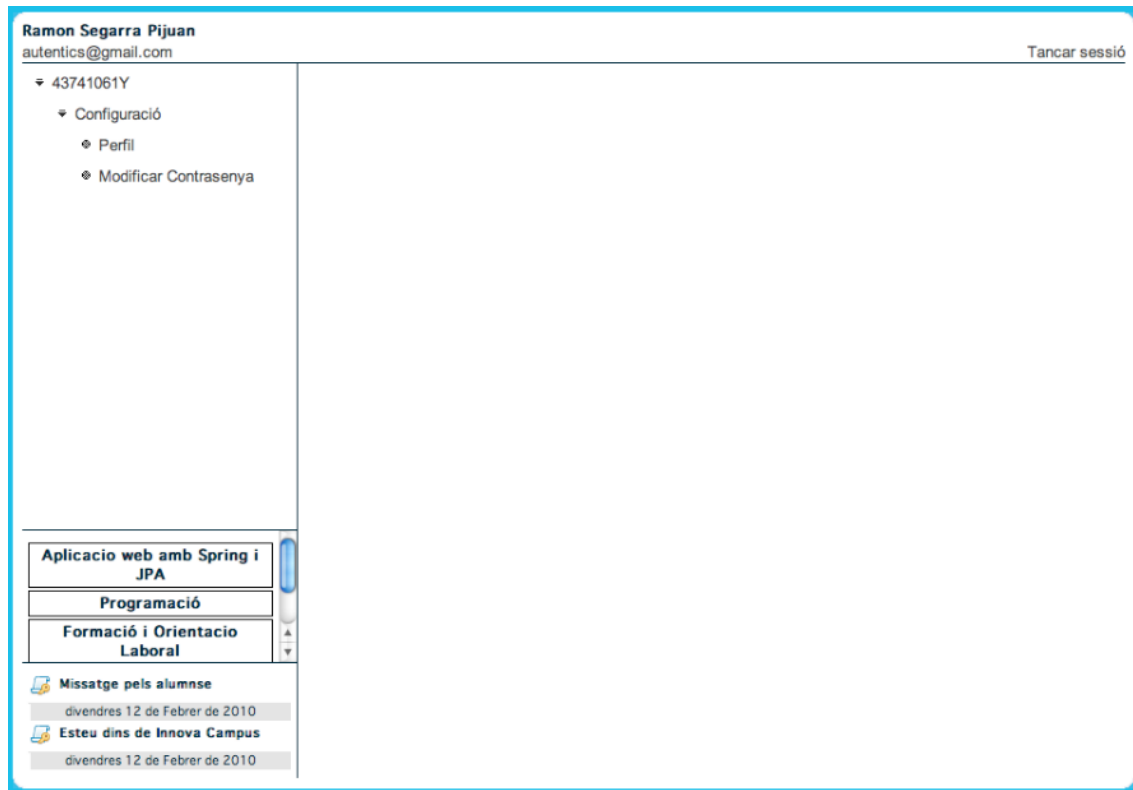


Figura 6.10: Pantalla de l'alumne

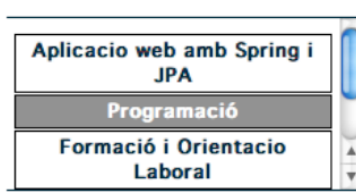


Figura 6.11: Menu d'assignatures

Al mateix temps que activem una assignatura, el menú en forma d'arbre modifica les seves opcions tenint en compte l'assignatura seleccionada.

Pantalla d'usuari professor

El funcionament de la pantalla del professor és la mateixa que la del alumne, es divideix en quatre parts, menú d'opcions en arbre, llista d'assignatures com a professor, anuncis i pantalla de continguts central.

Martin Pollinet
ramon.sepi@gmail.com

Tancar sessió

87654321Z - Programació

- Professors
- Alumnes
- Temes
- Objectius
 - Afegir
 - Consultar
- Preguntes
 - Afegir
 - Consultar
 - Importar Preguntes
 - Exportar Preguntes
- Respostes
- Anuncis
- Estadístiques

Català
Castellà
Hardware
Aplicacio web amb Spring i JPA

Benvinguts a tots !!!

Llista de preguntes

Tema: **Algoritmes** Objectius: **Teoria d'algoritme**

Text qüestió	Dificultat	T. mig	T. resp
Què és un algoritme ?	1.0	0	0
Quins valors podem emmagatzemar a les següent variables ? <code>int a,b,c,d;</code>	1.0	0	0
Que farà el següent codi ? <code>a:=b+c</code>	1.0	0	0
Respostes - Pregunta num: 24			
Concatenar el caràcter 'a' amb el 'b'			
Sumar dues variables			
Quin es el resultat correcte ? <code>int a,b:=2,c:=8</code> <code>a:=((a+b) *c) /2</code>	1.0	0	0
Què és un condicional ?	1.0	0	0

Avis legal | Sobre InnovaCampus

Figura 6.12: Pantalla del professor

Pantalla d'administrador

Pel que fa a la pantalla de l'administrador aquesta ha estat organitzada en dues parts, el menu d'opcions i la part central de contingut.

The screenshot displays the Administrator interface. On the left is a sidebar menu with the following items: 'admin' (selected), 'Configuració', 'Perfil', 'Modificar Contrasenya', 'Professors', 'Assignatures', and 'Anuncis'. The main content area is titled 'Configuració del perfil d'usuari' and contains a form with the following fields: 'DNI o NIE:' with the value 'admin', 'Nom:' with the value 'ADministrator', 'Cognoms:' (empty), and 'Correu electrònic:' with the value 'innovacampuss@gmail.com'. A 'Tancar sessió' link is located in the top right corner.

Figura 6.13: Pantalla de l'administrador

Capítol 7

Desenvolupament en comunitat

El programari lliure va néixer a principis dels anys vuitanta com un moviment per defensar i garantir les llibertats dels usuaris en el món de la informàtica. Davant d'un model industrial, cada cop més ferotge i restrictiu per als usuaris, el programari lliure es presenta com una alternativa molt més flexible per a la comunitat de desenvolupadors de software.

El fet que qualsevol persona pugui desenvolupar, utilitzar, distribuir i modificar programari lliure crea un context molt diferent en relació amb el del programari privatiu pel que fa als aspectes de desenvolupament, distribució de millores, coneixement, bloqueig tecnològic i cost de llicència. Tot plegat esdevé un marc ideal per al desenvolupament de software en comunitat.

7.1 Desenvolupament en comunitat i llicències

El programari lliure, generalment, es desenvolupa utilitzant metodologies i eines específiques, diferents de les que utilitzen els desenvolupadors de programari privatiu. El tret diferenciador del programari lliure és que es desenvolupa en comunitats formades per programadors —vinculats o no vinculats a empreses— interessats en el projecte.

Les comunitats de desenvolupadors de programari lliure s'organitzen al voltant d'un programari concret, formant grups de treball ben liderats i amb una distribució de tasques altament definida. Aquestes comunitats de desenvolupament utilitzen metodologies de treball basades en el treball en grup i la reutilització de la feina feta per altres projectes de programari lliure.

Els programes desenvolupats estan disponibles a la xarxa amb llicències que en permeten la modificació i distribució i que inclouen una descripció detallada de les condicions en què es pot fer.

En les comunitats de desenvolupament de programari lliure es poden diferenciar tres grups o rols:

- Nucli: grup reduït de persones que aporten la idea, defineixen tècnicament el projecte i n'inicien el desenvolupament. El nucli s'encarrega també de la coordinació i cohesió de la comunitat, així com de liderar la promoció o el desenvolupament del projecte.
- Codesenvolupadors: grup de persones que aporten el codi al projecte, pensen noves funcionalitats i busquen solucions a possibles problemes. Entre altres tasques, s'encarreguen d'aportar versions

per a plataformes diferents, de fer les traduccions de l'aplicació, etc.

- Usuaris actius: grup nombrós de persones que utilitzen el programa, reporten possibles errors, fan suggeriments de noves funcionalitats i participen en els fòrums de la comunitat de forma activa, etc.

La comunitat de desenvolupadors vetlla no només per la millora tècnica del producte, sinó que també s'encarrega d'impulsar-ne la difusió a la resta de la comunitat perquè tothom pugui treure'n el màxim profit.

Tan important és la llicència d'un programa lliure com la comunitat que l'ha desenvolupat i la manera com ha estat desenvolupat. Les comunitats de desenvolupament determinen la metodologia de treball, les eines de desenvolupament i la capacitat de fer evolucionar els mateixos programes.

Les comunitats de programari lliure ens ensenyen una manera d'organitzar-nos i treballar, en què la participació i la col.laboració, segons les possibilitats de cada persona, són molt importants i l'adaptació de la feina feta permet créixer i evolucionar ràpidament.

Si la comunitat de desenvolupament és activa, tindrem un producte que evoluciona constantment, amb les noves funcionalitats que aporta la comunitat, que podrem incorporar per millorar l'aplicació.

L'existència del programari inevitablement planteja la qüestió de com s'han de prendre les decisions sobre el seu ús. Per exemple, suposem que una persona que té una còpia d'un programa es troba amb una altra persona que en vol una. Aquestes persones tenen la possibilitat de copiar el programa:

- Qui ha de decidir si això es fa?
- Les persones implicades?
- O una tercera persona, anomenada "propietari"?

Quan els desenvolupadors de programari prenen en consideració aquestes preguntes parteixen de la suposició que el criteri per a la resposta és maximitzar els beneficis dels desenvolupadors. El poder polític de les empreses ha fet que el govern adoptés tant aquest criteri com la resposta proposada pels desenvolupadors: que el programa té un propietari, generalment una empresa relacionada amb elseu desenvolupament.

Podríem estudiar la mateixa pregunta a partir d'un criteri diferent: el de la prosperitat i la llibertat de la societat en general.

Aquesta resposta no es pot decidir amb la legislació actual, ja que la llei hauria de sotmetre's a l'ètica, no pas al contrari. La pràctica actual tampoc no respon a la qüestió, tot i que pot oferir respostes possibles. L'única manera de judicar-la és estudiant qui es beneficia i qui surt perjudicat del fet de reconèixer que el programari té propietaris, per què i en quina mesura. En altres paraules, s'hauria de fer una anàlisi de costos i beneficis en nom del conjunt de la societat, tenint en compte la llibertat individual i la producció de béns materials.

Així doncs, al nostre entendre els programadors tenen el deure d'animar la resta d'usuaris a compartir, redistribuir, estudiar i millorar el programari que escriuen: dit d'una altra manera, tenen el deure d'escriure programari lliure, i això és el que hem fet amb el projecte InnovaCampus.

El fet que InnovaCampus esdevingui una aplicació de programari lliure aporta un seguit de beneficis que resumirem a continuació:

- El desenvolupament. En el món del programari privatiu el desenvolupament del programari és dut a terme exclusivament pel fabricant. De vegades, però, les aplicacions poden ser esteses per mitjà de llenguatges script o d'interfícies de programació. Amb el programari lliure, com que es disposa d'accés al codi font i és possible modificar-lo, podem participar en el desenvolupament del producte, i incorporar-hi les funcionalitats que necessitem o personalitzar-lo tant com el projecte exigeixi. Addicionalment, molts competidors col·laboren en la feina de base de creació de producte, com en el cas de l'escriptori GNOME, que després personalitzen i comercialitzen com a part dels seus productes. Aquest treball col·laboratiu es tradueix en un reaprofitament dels esforços de terceres persones i, per tant, en una reducció de les inversions necessàries en desenvolupament i del temps necessari per completar el projecte.
- La distribució de millores. El programari lliure no només permet estendre un programa, sinó que a més ens dona dret a poder-ne distribuir les versions millorades. Podem fer una proposta de valor afegit sobre un programa ja existent i distribuir-lo, és a dir, construir les nostres propostes sobre una feina ja existent, cosa que redueix el temps i cost de desenvolupament.
- El coneixement. En el cas del programari privatiu, el coneixement de com està dissenyat i codificat el programa es troba únicament a mans del fabricant, ja que només ell disposa d'accés al codi font. Aquesta exclusivitat en l'accés al coneixement crea un monopoli de facto sobre l'aplicació, ja que, com dèiem, només el fabricant coneix en profunditat el funcionament intern del programa. En el món del programari lliure, com que es garanteix l'accés al codi font, qualsevol persona amb les habilitats i el temps suficient pot esdevenir un expert de primera línia en una aplicació.
- El bloqueig tecnològic. La possibilitat que ofereix el programari lliure d'accedir al codi font permet d'escriure filtres d'importació a pràcticament qualsevol format de dades conegut i d'adaptar l'aplicació a qualsevol plataforma nova (portabilitat d'aplicacions), cosa que minimitza el risc de bloqueig tecnològic.
- El Cost de llicència. Malgrat que ningú impedeix cobrar pel programari lliure, el cost tendeix a ser nul o menyspreable, en contrast amb el programari privatiu, que acostuma a tenir un cost elevat. Així doncs, el programari lliure permet tenir costos molt més baixos i solucions més competitives.

Tot el projecte InnovaCampus ha estat implementat i continuarà implementat-se sota la Llicència GPL (vegeu [Annex F]).

7.2 El desenvolupament d’InnovaCampus

Des de els seus orígens, l’aplicació Web d’innovació docent InnovaCampus, ha estat una projecte de programari lliure pensat per a ser desenvolupat en comunitat.

Tal i com hem avançat al llarg del capítol 1, el desenvolupament d’InnovaCampus ha passat per diferents etapes, des de la construcció d’un prototipus inicial fins a l’aplicació actual. La darrera etapa d’aquest prometedori projecte, consisteix en el redisseny complet de l’aplicació i s’ha dut a terme en dues fases ben diferenciades.

La primera fase va consistir en el desenvolupament de les capes de persistència i de negoci i fou implementada per en Xavier Aiguabella Guilera i la Natàlia Sans Solsona com a treball de final de carrera del segon cicle en Enginyeria informàtica a la Universitat de Lleida. (Vegeu [4])

La segona fase és la que dóna lloc al present treball de final de carrera i ha estat desenvolupada per en Ramon Segarra Pijuan i en Marcel·lí Alet Alís. Aquesta segona fase ha consistit en el disseny, implementació i integració de la capa de presentació.

Una de les premisses bàsiques del redisseny d’InnovaCampus ha estat la de dur a terme un disseny elegant i modular seguint un model enfocat al desenvolupament en comunitat. Per tal d’aconseguir-ho s’ha estat necessari:

- Documentació completa per desplegar l’aplicació (Vegeu [Annex E])
- Especificació de totes les classes de l’aplicació (en el cas d’InnovaCampus Javadoc de totes les classes)
- Una font de programari o repositori per tal de tenir un lloc centralitzat on emmagatzemar i mantenir el paquet de programari.
- Control de versions per tal de gestionar l’estat de l’aplicatiu en un moment determinat del seu desenvolupament i per permetre les contribucions concurrents al codi amb detecció de conflictes.

7.2.1 Eines emprades

Tal i com hem avançat en el punt anterior, les infraestructures que es requereixen per tal de desenvolupar correctament un projecte en comunitat són:

- Un repositori, en el nostre cas Sedna (<http://sedna.udl.cat/svnInnovaGWT>).
- Un gestor de versions, d’entre els que hem escollit Subversion.

A més a més, per tal de facilitar el desplegament de l’aplicació hem creat el corresponent script ANT per tal de procedir amb el desplegament de l’aplicació i permetre una ràpida i flexible adaptació als IDE’s Java més comuns com Eclipse i NetBeans.

Per a més informació sobre el repositori Sedna consultar [Annex A] i [14]. Per a més informació sobre el gestor de versions Subversion consultar [Annex A] i [15]. Per consultar el manual de desplegament veure [Annex E].

7.3 Planificació del projecte

La planificació del projecte ens clarifica l'ordre de les tasques i realitza una estimació del temps que es necessita per realitzar-les. És en aquesta fase on ens hem de donar compte de si el projecte és excessivament complex o si és assequible. Llavors es decideix si s'amplia o es redueix el projecte. La planificació es realitza amb sis passos, que requereixen el domini de certes tècniques.

1. Divisió del treball
2. Estimació dels temps
3. Identificació de les fites (valoracions)
4. Encadenament d'activitats
5. Planificació temporal
6. Replanificació

Hem utilitzat un programa estàndard de gestió de projectes per a realitzar millor la planificació del projecte. En la planificació i consten totes les fases del projecte, la seva duració i si depenen entre elles o no.

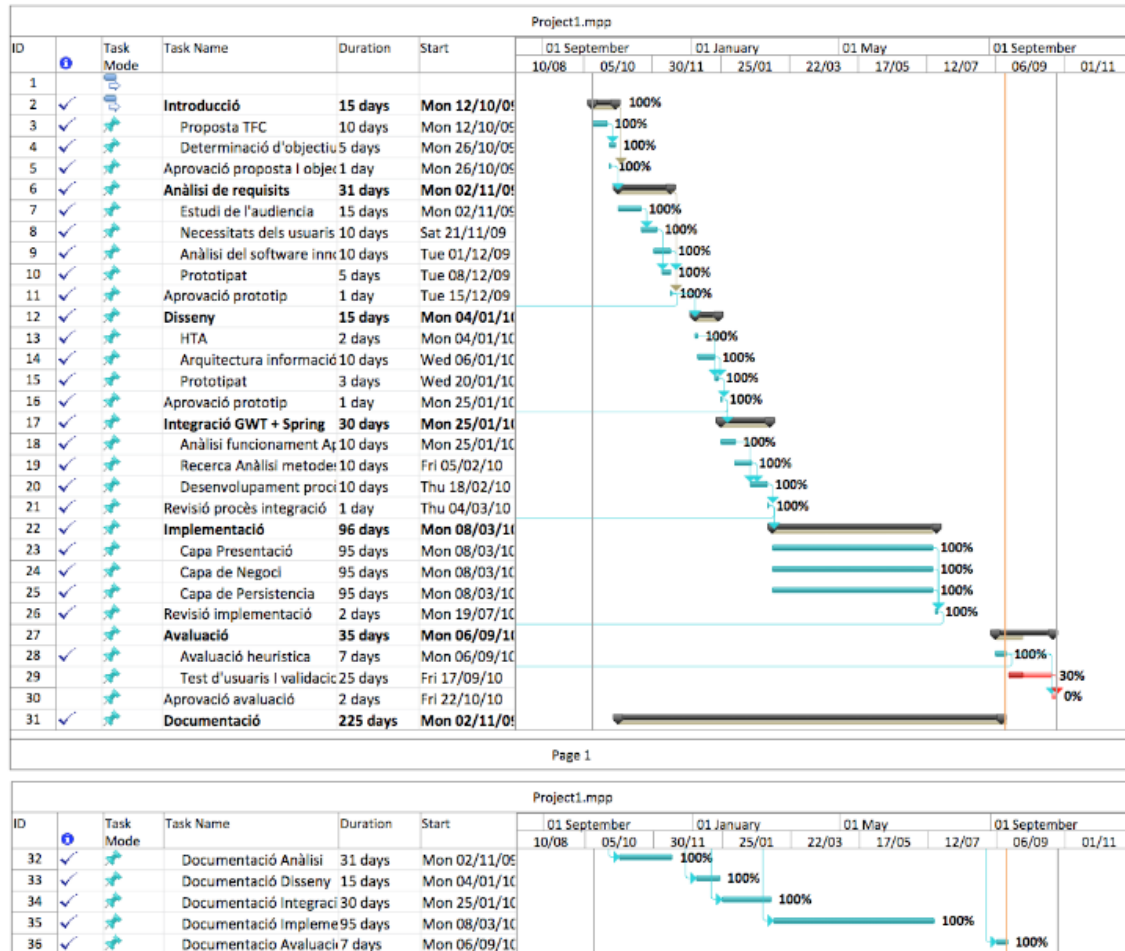


Figura 7.1: Diagrama de Gantt de les tasques del projecte

Etapas:

Tal i com podem observar a la figura anterior, el projecte es divideix en diferents etapes on cadascuna de elles té una propòsit diferent. A continuació detallem cada una de les etapes que conformen la totalitat del projecte.

1. Introducció

- (a) Presentació i aprovació proposta TFC
- (b) Determinar els objectius principals del projecte

2. Anàlisi de requisits

- (a) Estudi de l'audiència (els futurs usuaris)
- (b) Necessitats dels usuaris potencials de l'aplicació
- (c) Anàlisis del software InnovaCampus 1.0
- (d) Desenvolupament d'un primer prototip de software (en paper)

3. Disseny

- (a) Anàlisis jeràrquic de tasques (Passos per realitzar una tasca)
- (b) Arquitectura de la informació (Organització del contingut)
- (c) Desenvolupament d'un segon prototip de software (en digital)

4. Integració de tecnologies (GWT+SPRING Framework)

- (a) Estudi del funcionament del software InnovaCampus 1.0
- (b) Recerca del mètode més adient d'integració amb l'anterior tecnologia
- (c) Desenvolupament del procés d'integració de tecnologies (velles amb noves)

5. Implementació

- (a) Desenvolupament de la capa presentació (Interfície)
- (b) Desenvolupament de la capa negoci (Lògica de processament)
- (c) Adaptacions de la capa persistència

6. Avaluació

- (a) Avaluacions, validacions de processos i tests d'usuaris

7. Documentació de cada una de les etapes del projecte

Fites:

Durant el desenvolupament del projecte es realitzen una serie de reunions, valoracions i aprovacions per tal de poder controlar el ritme de treball i que la feina de cada subetapa segueix dins les dates fixades, per tal de que no es descompensi la temporalitat global del projecte, ja que si una etapa s'allarga molt més del temps previst, el projecte global se'n veu afectat ja que existeixen moltes etapes, on no es pot iniciar si no s'ha finalitzat l'anterior. Pel que fa al nostre projecte aquestes fites estan situades al finalitzar cada una de les etapes que hem descrit anteriorment. Per tant no es pot començar amb una etapa nova si la seva predecessora no ha estat aprovada.



Figura 7.2: Temporalitat total del projecte

Capítol 8

Conclusions i treball futur

En aquest projecte hem dut a terme el disseny, integració i desenvolupament d'una nova capa de presentació per a l'aplicació d'innovació docent InnovaCampus. Partint de la feina feta pels nostres companys Xavier Aiguabella i Natàlia Sans en una primera fase del projecte, la qual pretenia desenvolupar les capes de persistència i de negoci, hem plantejat una solució a la capa de presentació basada en el nou framework de Google, conegut com a Google Web Toolkit.

L'elecció de Google Web Toolkit fou una decisió força arriscada, doncs no únicament calia desenvolupar la interfície gràfica, sinó que també calia donar resposta a la integració d'aquest nou framework amb les tecnologies emprades pels nostres companys en el disseny de la resta de capes (Spring MVC+JPA). L'inici del projecte fou força dur, doncs cal tenir en compte que la documentació existent per tal d'integrar aquestes tecnologies és més que escassa, no obstant, finalment, i després de moltes deliberacions, varem aconseguir dissenyar una solució elegant com a resposta a la integració d'aquestes tecnologies.

Un cop integrades les tecnologies, el següent pas fou el de treure el màxim profit a Google Web Toolkit. Es tracta d'un framework que ens ha sorprès molt gratament, doncs, des de el nostre punt de vista, en certa manera reinventa la forma de programar aplicacions Web. Aquesta nova forma de programació resol molts dels problemes que ofereixen les pàgines Web tradicionals, i que amb l'aparició d'AJAX s'han fet encara més notables. La solució és molt bona per a la nostra comunitat, la dels programadors d'aplicacions, doncs ens estalvia el fet d'haver de barallar-nos constantment amb la tecnologia HTML/JavaScript, que originalment no fou dissenyada per al desenvolupament d'aplicacions Web interactives.

Deixant de banda les tecnologies emprades, també volem destacar que aquest projecte també ens ha fet partíceps del món del desenvolupament en comunitat i ha enriquit la nostra experiència relacionada amb el desenvolupament de programari lliure. Així doncs, hem continuat la tasca iniciada per altres desenvolupadors i, d'una forma metòdica i col·laborativa, emprant gestors de versions, documentant les classes implementades i elaborant tota la documentació necessària per tal de facilitar la tasca als futurs desenvolupadors de l'aplicació, hem contribuït en el disseny d'un entorn còmode que permeti a la comunitat familiaritzar-se ràpidament amb el context del projecte.

En general i com a conclusió a nivell personal, destaquem l'aprofitament dels coneixements adquirits durant la titulació, els quals han esdevingut fonamentals en les diferents fases del projecte. També cal destacar que hem enriquit els nostres coneixements amb recursos tecnològics nous per a nosaltres i que ens han sorprès gratament. El món de la informàtica, és doncs, tan extens que ens obliga a

adaptar-nos i actualitzar-nos, dia a dia, a noves fonts de recursos, els quals permeten l'elaboració de noves i millors aplicacions. Així doncs, al llarg d'aquest projecte hem pogut enriquir considerablement els nostres coneixements gràcies al contacte amb aquestes tecnologies tan actuals i innovadores, hem aprofundit en nous àmbits que ens han ofert un gran ventall de possibilitats que han permès donar un valor afegit a aquesta eina d'autoavaluació.

Finalment, i seguint amb la tònica d'optimisme i satisfacció que ens aporta haver finalitzat amb èxit bona part dels objectius plantejats, volem fer èmfasi en el fet que la nostra tasca no acaba aquí, doncs, hem desenvolupat una aplicació però ens queda donar resposta a la posada en marxa de la mateixa i al suport de futures incidències. A més a més, com a a aspectes a millorar i treball futur creiem que seria interessant:

- Analitzar la creació de nous tipus de tests interactius fent ús de les noves possibilitats que aporten les tecnologies emprades.
- Oferir noves funcionalitats tant a estudiants com a professors per tal de fer l'aplicació més personalitzable i que permetin embolcallar-la amb noves possibilitats que enriqueixin, encara més, l'aplicació.
- Oferir un marc de treball específic per a matemàtics que permeti introduir fórmules sense problemes, tot i que l'aplicació actual ja ha suposat un avenç en aquest sentit.
- Oferir un nou marc enfocat a la introducció interactiva de tests enfocat a assignatures de programació.
- Plantejar un subconjunt d'accions disponibles en mode desconnectat.
- Plantejar millores en l'accessibilitat a persones discapacitades.
- Auditoria i registre d'accions per millorar la seguretat de l'aplicació.

Apèndix A

Altres tecnologies emprades

La tecnologia més rellevant sota la qual s'ha desenvolupat aquest projecte és ha estat el framework GWT. Aquesta tecnologia s'ha tractat de forma extensa i detallada en el capítol 4 de la memòria. Veurem ara altres tecnologies utilitzades en el desenvolupament de l'aplicació com són el llenguatge de marques HTML o el llenguatge interpretat JavaScript.

A.1 El llenguatge HTML

HTML (Acrònim de Hyper Text Markup Language, en català, "llenguatge de marques d'hipertext"), és un llenguatge de marques que deriva de l'SGML dissenyat per estructurar textos i relacionar-los en forma d'hipertext. Gràcies a Internet i als navegadors web, s'ha convertit en un dels formats més populars que existeixen per a la construcció de documents. Es tracta, però, d'un llenguatge que genera pàgines estàtiques, per això s'enriqueixen amb la utilització d'altres llenguatges com Javascript, el qual veurem a continuació.

Podeu trobar més informació a [8].

A.2 El llenguatge JavaScript

Javascript és un llenguatge interpretat orientat a les pàgines web, amb una sintaxis semblant a la del llenguatge Java. S'utilitza en les pàgines Web, per a realitzar tasques i operacions en el marc de l'aplicació client servidor.

Com ja hem dit, JavaScript és un llenguatge interpretat, és a dir, no requereix compilació. El navegador de l'usuari s'encarrega d'interpretar les sentències JavaScript contingudes en una pàgina HTML i executar-les adequadament.

JavaScript és un llenguatge orientat a esdeveniments. Quan un usuari clica sobre un enllaç o mou el punter sobre una imatge es produeix un esdeveniment. Mitjançant JavaScript es poden desenvolupar scripts que executin accions com a resposta a aquestos esdeveniments.

JavaScript és també un llenguatge orientat a objectes. El model d'objectes de JavaScript està reduït i simplificat, però inclou els elements necessaris per a què els scripts puguin accedir a la informació

d'una pàgina i puguin actuar sobre la interfície del navegador.

Fem una breu síntesi de les característiques més importants (les ja citades i algunes no citades) de JavaScript:

Característiques tècniques de JavaScript.

- Interpretat (no compilat) des de el navegador del client.
- Basat en objectes. Només els utilitza, manca d'herències i de classes.
- El codi està integrat en el HTML. Queda ocult a l'usuari, però pot formar part del mateix arxiu.
- No és necessari declarar el tipus de variables.
- Enllaçat dinàmic. Els objectes als que es fa referència han d'existir en temps d'execució (lenguatge interpretat).
- No requereix la presència de servidor, ja que manté un funcionament local.
- L'aplicació resideix en el client.

JavaScript a la nostra aplicació.

JavaScript ocupa un lloc força important dins de la nostra aplicació. Ens ha permès dotar de dinamisme les pàgines sense haver de dependre sempre del servidor. Tot seguit enumerarem les aportacions més importants de JavaScript a l'aplicació:

- El tractament de les pestanyes on es mostren les diferents assignatures que es poden escollir, així com el desplaçament per elles en el cas que no totes es mostrin per pantalla al mateix instant (quan hi ha més de 5 assignatures).
- Obertura de finestres emergents.
- Validació d'aquells aspectes en els quals no em cregut convenient aplicar filtres.
- Els testos que resolen els alumnes són presentats mitjançant JavaScript. D'aquesta manera podem aconseguir que les preguntes es mostrin una a una mitjançant esdeveniments sense haver d'estar enviant peticions al servidor contínuament. També ens permet controlar el temps que tenen els estudiants per a resoldre el test.

En els punts anteriorment citats hem vist algunes de les aportacions de l'ús de JavaScript en la nostra aplicació, així doncs, com en la majoria d'aplicacions de la xarxa, JavaScript també ha resultat una "eina" de gran utilitat en el desenvolupament del nostre software.

Podeu trobar més informació a [15].

A.3 El llenguatge CSS

Els fulls d'estil en cascada (Cascading Style Sheets, CSS) són un llenguatge formal usat per a definir la presentació d'un document estructurat escrit en HTML o XML (i per extensió en XHTML). El W3C (World Wide Web Consortium) és l'encarregat de formular l'especificació dels fulls d'estil que servirà d'estàndard per als agents d'usuari o navegadors. La idea que es troba darrere del desenvolupament

de CSS és separar l'estructura d'un document de la seva presentació.

Els avantatges d'utilitzar CSS (o altre llenguatge d'estil) són:

- Control centralitzat de la presentació d'un lloc web complet amb el que s'agilita de forma considerable l'actualització del mateix.
- Els navegadors permeten als usuaris especificar el seu propi full d'estil local que serà aplicada a un lloc web remot, amb el que augmenta considerablement l'accessibilitat. Per exemple, persones amb deficiències visuals poden configurar el seu propi full d'estil per a augmentar la grandària del text o remarcar més els enllaços.
- Una pàgina pot disposar de diferents fulles d'estil segons el dispositiu que la mostri o fins i tot a elecció de l'usuari. Per exemple, per a ser impresa, mostrada en un dispositiu mòbil, o ser "llegida" per un sintetitzador de veu.
- El document HTML en si mateix és més clar d'entendre i s'aconsegueix reduir considerablement la seva grandària.

A.4 El llenguatge XML

XML (Acrònim de eXtensible Markup Language, en català, "llenguatge de marques extensible"), és un llenguatge de marques que deriva de l'SGML i està desenvolupat per el World Wide Web Consortium (W3C).

El seu objectiu principal és aconseguir una pàgina Web més semàntica. Tot i que una de les principals funcions amb les que va néixer seria succeir a HTML, separant l'estructura del contingut i permetent el desenvolupament de vocabularis modulars, compatibles amb certa unitat i simplicitat del llenguatge (objectiu que es desenvolupa a través de l'especificació XHTML), té altres aplicacions entre les que destaca el seu ús com a estàndard per a l'intercanvi de dades entre diverses aplicacions o software.

De la mateixa manera que HTML, es basa en documents de text pla en els quals s'utilitzen etiquetes per delimitar els elements del document. En canvi, XML defineix aquestes etiquetes en funció del tipus de dades que està descrivint i no de l'aparença final que tindrà a la pantalla o a la còpia impresa, a més a més de permetre definir noves etiquetes i ampliar les ja existents.

Que un document estigui ben format no diu res sobre si la seva estructura és correcta. Per tant, calen mecanismes per tal de definir aquesta estructura. Actualment hi ha dos mecanismes per tal de fer-ho: DTD i XML Schema. En el nostre cas veurem el mecanisme DTD que és el que hem utilitzat.

DTD (Document Type Definition)

Un DTD és un conjunt de regles que defineixen les etiquetes, el seu ordre, la seva jerarquia, els seus atributs i els valors que aquests poden prendre. Prové directament de SGML, serveix per a definir la gramàtica d'un llenguatge definit sobre el metallenguatge XML (si un document XML compleix la seva gramàtica direm que és vàlid).

Podeu trobar més informació a [9].

A.5 El llenguatge JAVA

El llenguatge de programació Java fou dissenyat per James Gosling i els seus companys a Sun Microsystems, a l'any 1990, a partir del C++. Des del seu naixement fou pensat com un llenguatge orientat a objectes, és a dir, que segueix la filosofia de programar mòduls senzills, per tal de crear aplicacions avançades quan tots treballen junts.

Es tracta d'un llenguatge interpretat i, per tant, pot semblar lent en comparació amb altres llenguatges, però ofereix un índex de reutilització del codi molt elevat i amb un important nombre de llibreries de lliure utilització. És un llenguatge flexible i potent per la facilitat amb què es programa i els resultats que ofereix. Un dels trets que el caracteritza i que fa que sigui una eina molt valorada a l'hora de desenvolupar aplicacions distribuïdes, és el fet que sigui un llenguatge multi-plataforma.

Característiques bàsiques del llenguatge:

- Senzill: Java s'ha creat per a ser un llenguatge senzill i amb una elegant sintaxis. Només consta de tres tipus de dades primitives i ha eliminat els punters i l'herència múltiple.
- Orientat a objectes: Java segueix els paradigmes de la programació orientada a objectes, ja que la programació amb Java es centra en la manipulació, creació i construcció d'objectes
- Distribuït: Java permet la construcció d'aplicacions distribuïdes per mitjà d'una col·lecció específica de classes .
- Interpretat: Es necessita un intèrpret per executar els programes de Java, això els fa més lents però més flexibles
- Robust: Java és un llenguatge robust i fiable, s'ha escrit pensant en poder verificar errors i està molt tipificat.
- Segur: Java té escassos problemes de seguretat, característica molt important en les aplicacions distribuïdes per Internet.
- Arquitectura neutral: Java és independent de la plataforma final on s'executarà el programa.
- Portabilitat: Java és un llenguatge d'alt nivell i independent de la plataforma, això el fa portable.
- Alt rendiment: Els compiladors Java han anat millorant les seves prestacions al llarg del temps.
- Concurrent: Java permet l'execució de múltiples fils d'execució o varies tasques simultàniament.
- Dinàmic: En temps d'execució, l'entorn Java es pot estendre mitjançant enllaços a classes que poden estar localitzades en servidors remots o en una xarxa.

Per més informació [12], [13], [19], [21] i [22].

A.6 AJAX

AJAX són les sigles de Asynchronous Javascript And Xml, (JavaScript asíncron i XML), un conjunt de tecnologies que permeten actualitzar continguts web sense haver de tornar a carregar la pàgina. Això obre la porta a pàgines web interactives.

AJAX és asíncron en tant que les dades addicionals són demanades i carregades en un segon pla, sense interferir en la presentació i el comportament de la pàgina. Habitualment les funcions d'AJAX

es criden des del llenguatge JavaScript. Les dades són adquirides usant l'objecte XMLHttpRequest, que està disponible als llenguatges d'script que corren en navegadors moderns, o també usant Remote Scripting en navegador que no suportin XMLHttpRequest. En qualsevol cas, no és necessari que el contingut asíncron sigui formatat en XML.

AJAX és multiplataforma i es pot usar en diversos sistemes operatius, arquitectures de computador i navegadors web, ja que es basa en estàndards oberts com JavaScript i DOM. Hi ha implementacions open source de frameworks i llibreries.

Una possible tècnica per programar en AJAX és mitjançant un motor AJAX (escrit en JavaScript). Aquest motor és l'encarregat de mostrar la interfície gràfica i comunicar-se amb el servidor. Mitjançant AJAX, l'usuari no es connecta directament amb el servidor, sinó que primer ho fa amb el motor d'AJAX i aquest és el que fa la sol·licitud al servidor. Un cop això, el servidor retorna la sol·licitud que passa a través del motor i renderitza el resultat, de manera que mai es veurà la pàgina completament blanca com passa amb els models clàssics d'aplicacions web.

A.7 El llenguatge JSP

JSP és una tecnologia basada en Java que simplifica el desenvolupament d'aplicacions web amb contingut dinàmic. Amb JSP, els dissenyadors web poden incorporar elements dinàmics dins d'una pàgina utilitzant tant porcions de codi Java incrustades, com unes quantes etiquetes. Així, les pàgines JSP tenen l'aspecte d'una pàgina HTML tradicional, a la que s'ha introduït part de codi Java junt amb unes etiquetes. D'aquesta manera, quan una pàgina és sol·licitada per un usuari i processada per un servidor HTTP, el codi HTML passarà directament a l'usuari, mentre que les porcions de codi Java seran executades al servidor per tal de generar el contingut dinàmic de la pàgina. Quan l'usuari accedeixi al codi de la pàgina només veurà HTML, sense poder accedir, per tant, al codi JSP subjacent.

Evolució tecnològica:

El fet que una pàgina contingui contingut dinàmic, exigeix que el servidor web realitzi un treball de processament que en el cas de les pàgines estàtiques no és necessari. La forma de realitzar aquest processament ha anat canviant amb el temps.

Inicialment la generació de continguts dinàmics es feia fora dels servidors. Quan arribava una sol·licitud, aquesta era processada al servidor, i quan era necessari es cridava a un procés, fora del servidor, que generava el contingut dinàmic i el retornava al servidor. Els models basats en el Common Gateway Interface (CGI) seguien aquesta idea. Alguns dels seus principals problemes eren la necessitat d'overhead per a la comunicació entre procés i servidor o el consum de recursos del sistema per part del procés.

Posteriorment es va optar per sistemes que introduïen porcions de codi de llenguatges tradicionals incrustats dins de la pàgina utilitzant etiquetes. Aquest model l'han seguit entre d'altres: Microsoft Active Server Pages (ASP), Server Side JavaScript (SSJS), Java Server Pages (JSP) i altres.

Pel que fa a JSP, el llenguatge que s'utilitza per a la generació del contingut dinàmic és, típicament, Java, i proveeix un conjunt d'etiquetes que interactuen amb objectes Java al servidor de manera que no és estrictament necessari que aparegui codi Java a la pàgina.

Beneficis:

JSP ofereix varis beneficis com a sistema de generació de contingut dinàmic. Al estar basat en Java, presenta les avantatges que aquest llenguatge ofereix amb respecte a la portabilitat entre plataformes i les derivades de l'orientació a objectes d'aquest llenguatge.

Realització:

Les peticions de pàgines JSP són normalment implementades mitjançant Servlets, de forma que el contenidor de Servlets (Tomcat), maneja múltiples sol·licituds a la vegada, requerint un menor overhead, i per tant requerint menys recursos. Això fa que JSP sigui molt més eficient que altres models com per exemple els programes CGI.

Components reutilitzables:

Aquesta característica deriva de l'orientació a objectes de Java. JSP permet implementar contingut dinàmic incloent directament codi Java a la pàgina, però també ofereix una sèrie d'etiquetes que permeten actuar sobre objectes Java residents al servidor. Aquests objectes es comporten com a caixes negres a les que la pàgina accedeix sense tenir coneixement de com funcionen internament, i per tant, alliberant al creador de la pàgina de la programació. Aquest objectes, a més a més, podran ser reutilitzats amb el simple fet de conèixer la seva funcionalitat.

Separació de presentació i implementació:

Aquesta avantatge prové directament de l'anterior. El fet que la implementació del programa pugui ser realitzada pels objectes Java, permet separar el que és la presentació en si, i el codi encarregat de generar la informació necessària que apareixerà a la pàgina.

Divisió de la tasca:

La separació de presentació i implementació permet deslligar les tasques encarregades de desenvolupar ambdós aspectes. Així doncs, algú que no sàpiga res de Java, podria encarregar-se de la part de la pàgina relacionada amb la presentació, li seria suficient conèixer les propietats que li ofereix un conjunt d'objectes, i així, accedint a ells aconseguiria la informació necessària. De la mateixa manera, un programador Java, seguint una sèrie de normes a l'hora de crear els objectes, s'encarregaria de crear el codi que generarà la informació dinàmica, despreocupant-se dels problemes de presentació de la pàgina.

Podeu trobar més informació relacionada amb la tecnologia JSP a [5], [6], [9] i [10].

A.7.1 El servlet

La majoria de les implementacions de JSP estan basades en els Servlets. Per tant, per tal de comprendre com treballa una pàgina JSP, primer hem de comprendre com treballen els Servlets.

Els Servlets són programes basats en Java, anàlegs als programes CGI, implementats mitjançant un contenidor de Servlets associats a un servidor HTTP. El fonament dels Servlets és el següent: un conjunt de URLs són configurades per a ser administrades per el contenidor de Servlets, de manera que sempre que arribi una sol·licitud per a una d'aquestes URLs al servidor, aquest l'envia al contenidor de Servlets per a que el processi.

La forma d'enviar-lo és creant un objecte Java que empaqueti totes les dades de la sol·licitud. Un objecte Java també és creat representant la resposta. Ambdós objectes tindran els seus mètodes d'accés, d'aquesta forma, el contenidor de Servlets accedeix a les dades de la sol·licitud per tal de realitzar les operacions necessàries sobre els mateixos i així construir la resposta. El codi HTML generat com a resposta (no s'ha d'oblidar que el contingut dinàmic generat és codi HTML) és escrit a la cadena de sortida associada a l'objecte resposta, i aquest objecte és enviat al servidor HTTP, el qual la retorna al navegador que havia fet la sol·licitud en primer lloc. En el cas que existeixin múltiples sol·licituds per a un Servlet, aquestes són administrades executant cada crida als mètodes dels Servlets en diferents fils d'execució.

A.7.2 Java Server Pages

El component principal d'una implementació de JSP basada en Servlets és un Servlet especial anomenat compilador de pàgina. El contenidor està configurat per a cridar a aquest Servlet sempre que arribi una sol·licitud a una pàgina JSP. És aquest compilador de pàgina i la seva classe Java associada el que converteix el contenidor Servlet en un contenidor JSP.

El procediment és el següent: Quan arriba al servidor HTTP una sol·licitud d'una pàgina JSP, aquesta és enviada al contenidor JSP, el qual invoca al compilador de pàgina per a que s'encarregui de la mateixa. El compilador analitza el contingut de la pàgina buscant etiquetes JSP, traduint el seu contingut en el codi Java equivalent que, al ser executat, generarà el contingut dinàmic. Mesclant el contingut estàtic de la pàgina original junt amb el codi Java del contingut dinàmic, es generarà un Servlet amb els seus mètodes de servei. Una vegada que tot el codi del Servlet hagi estat construït, el compilador de la pàgina crida al compilador Java per a compilar aquest codi i afegir el fitxer de classe Java resultant al directori apropiat en el path de les classes del contenidor JSP. Tot aquest procés només es realitza la primera vegada que es sol·licita una pàgina JSP, la resta de sol·licituds són remeses directament al Servlet compilat. Així quan es crida a una pàgina JSP el compilador de pàgina invoca a aquest Servlet per tal de generar la resposta per a la sol·licitud original.

Resumint, podríem dir que les sol·licituds del navegador arriben al servidor HTTP i les pàgines JSP són enviades al Servlet compilador de pàgines que corre al contenidor JSP. Si el Servlet per a la pàgina actual està actualitzat el genera i compila, carregant-lo al contenidor de Servlets. En el cas contrari el control és transferit al Servlet de la pàgina JSP que s'encarrega de manejar la sol·licitud generant la resposta i enviant-la al servidor HTTP el qual la remetrà al navegador.

Podeu trobar més informació sobre la tecnologia JSP a [5], [6], [7] i [10].

A.8 Spring Framework

L'Spring framework (abreviant, Spring), és un marc de treball de codi obert per la plataforma Java. La primera versió va ser escrita per Rod Johnson, que inicialment va llançar el producte juntament amb el llibre *Expert One-on-One Java EE Design and Development* (Wrox Press, octubre 2002). També hi ha un port disponible per la plataforma .NET, Spring.NET.

El marc de treball va ser inicialment llençat al juny de 2003 sota la llicència Apache 2.0. La primera versió major 1.0 va ser distribuïda el març de [[2004], amb llançaments addicionals el setembre de 2004 i març de 2005.

Encara que Spring Framework no força cap model de programació, ha esdevingut àmpliament popular dintre de la comunitat Java primerament com una alternativa que desplaçaria el model Enterprise JavaBean. Per disseny aquest entorn ofereix una gran llibertat als desenvolupadors de Java i a més proveeix solucions fàcils i ben documentades per pràctiques habituals en la indústria.

Mentre les funcionalitats del nucli de l'entorn són usables en una aplicació Java hi ha diferents extensions i millores per construir aplicacions web damunt d'una plataforma Java EE. Gràcies a això, Spring ha aconseguit una gran popularitat i és reconegut pels fabricants com un entorn estratègicament important.

Funcionalitats clau:

- Gestió de la configuració basada en JavaBeans, aplicant-hi principis d'Inversió de Control, més específicament usant la tècnica d'Injecció de Dependència. Això ajuda a reduir les dependències de components, en implementacions específiques, sobre altres components.
- Una factoria de Beans central, que és usada globalment.
- Capa genèrica d'abstracció per la gestió de transaccions de la base de dades.
- Estratègies preincorporades per la JTA i un sol DataSource de JDBC. Això elimina la dependència en un entorn Java EE pel suport a les transaccions.
- Integració amb entorns de persistència com Hibernate, JDO, iBatis i db4o.
- Entorn d'aplicació web MVC, construït al nucli de la funcionalitat de Spring. suportant moltes tecnologies per generar vistes, incloent-hi JSP, FreeMaker, Velocity, Tiles, iText i POI.
- Entorn extensiu de programació orientada a aspectes per proveir serveis, com ara gestió de les transaccions. Amb això es millora la modularitat dels sistemes.

Podeu trobar més informació sobre Spring a [26], [27] i [28].

A.9 JPA

Java Persistence API (JPA) és un marc de treball per a Java que permet als programadors gestionar dades relacionals en aplicacions que fan servir la plataforma Java.

La persistència consisteix en tres àrees:

- L'API, definida al paquet `javax.persistence`.
- El Java Persistence Query Language.
- Les metadades objecte/relacional.

Per més informació [32].

A.10 SVN

Subversion (SVN) és un sistema de control de versions de codi obert. Gestiona fitxers i directoris, i els canvis que es produeixen en ells en el temps. Això li permet recuperar versions antigues de les dades o controlar l'evolució de canvis de les dades.

En aquest sentit, moltes persones comparen un sistema de control de versions com una "màquina del temps".

Característiques principals:

- Se segueix la història dels fitxers i directoris a través de còpies i reanomenats.
- Les modificacions (incloent canvis a diversos fitxers) són atòmiques.
- Es pot servir mitjançant Servidor HTTP Apache sobre WebDAV/DeltaV.
- Gestiona eficientment fitxers binaris.

Per més informació [30].

Apèndix B

La capa de persistència

La capa de persistència d'una aplicació és l'encarregada d'emmagatzemar les dades i facilitar l'accés a elles. Aquesta capa pot estar formada per un o més gestors de bases de dades que realitzen tot l'emmagatzemament de dades, reben les sol·licituds d'emmagatzemament i recuperació d'informació des de la capa de negoci.

En aquesta aplicació el gestor és el PostgreSQL, i les classes que interactuen directament amb les dades són les classes DAO (Data Access Object) implementades en JPA (Java Persistence API).

El punt 2.6 del capítol 2 ens ha presentat el disseny actual de la base de dades actual d'InnovaCampus, no obstant, el present annex pretén oferir una aproximació més tècnica a la implementació de la capa de persistència.

B.1 ORM: JPA

En el redisseny d'InnovaCampus s'ha optat per fer un mapeig de les classes del domini mitjançant Object Relational Mapping(ORM) i implementant-ho utilitzant el patró de Data Access Object (DAO). En aquesta secció explicarem en que consisteix ORM i després exposarem el patró DAO.

ORM es basa en desenvolupar codi Java i mapejar-lo a la base de dades. Es poden utilitzar diferents tecnologies, com Hibernate o JPA. En aquest cas s'ha emprat JPA.

Per veure bé com s'utilitza ORM i més concretament JPA explicarem el concepte d'entitats i com s'interrelacionen entre elles, com Spring i JPA interaccionen mitjançant EntityManager, i finalment el Java Persistence Query Language (JPQL) utilitzat en JPA.

Entitats i relacions:

Cada interfície DAO disposa d'una implementació en JPA. Java Persistence API (JPA) està formada per:

- L'API de persistència.
- El Java Persistence Query Language (JPQL).
- Les metadades objecte/relació.

En primer lloc cal implementar les entitats mitjançant classes lleugeres en Java, que acostumen a representar una taula de la base de dades, i cadascuna d'aquestes ha de contenir els atributs de la taula. Davant de cada atribut s'han d'inserir les metadades corresponents per indicar quin tipus d'atribut és i quines relacions té amb atributs d'altres taules. Això també pot indicar-se a través d'un arxiu XML. A InnovaCampus s'han utilitzat les classes de domini com a entitats.

Un cop està tot enllaçat ja es pot començar a desenvolupar codi en JPQL o operacions definides per defecte en JPA.

Seguidament mostrarem i explicarem el codi desenvolupat per a la persistència d'estudiants.

```
@Entity
@Table(name="USERS")
@Inheritance(strategy=InheritanceType.JOINED)
@DiscriminatorColumn(name="ROLE", discriminatorType=DiscriminatorType.STRING, length=1)

public class User implements Serializable {

    @Id
    private String NIF;
    @Column(name="PASSWORD", nullable=false)
    private String Password;
    @Column(name="EMAIL", nullable=true)
    private String Email;
    @Column(name="NAME", nullable=true)
    private String Name;
    @Column(name="SURNAMES", nullable=false)
    private String Surnames;
    @Column(name="ROLE", nullable=false)
    private char Role;

    [...]
}
```

En primer lloc s'indica que es tracta d'una entitat mitjançant @Entity, seguidament la taula a la que fa referència amb @Table i el nom d'aquesta, s'especifica que forma part d'una jerarquia amb @Inheritance, i l'atribut que actua com a discriminador de la jerarquia amb @DiscriminatorColumn.

Finalment a cada camp de la classe s'hi col·loca l'etiqueta @Column per relacionar-lo amb la columna de la taula de la base de dades. En el cas de la clau primària s'utilitza @Id, i no s'indica a quin atribut fa referència perquè utilitza el mateix nom (tenint en compte que Java és case sensitive).

```

@Entity
@Table(name="STUDENT")
@DiscriminatorValue(value="S")
public class Student extends User implements Serializable
{

    @OneToMany(mappedBy="student",cascade=CascadeType.REMOVE)
    private List<Test> tests = new LinkedList();
    @ManyToMany
    @JoinTable(name="ENROLMENT",
    joinColumns = @JoinColumn(name="NIF",referencedColumnName="NIF"),
    inverseJoinColumns = @JoinColumn(name="SUBJECT_ID",referencedColumnName="ID"))
    private List<Subject> subjects = new LinkedList();

    [...]
}

```

A banda del que s'ha explicat a l'exemple User.java aquí pot observar-se l'etiqueta @DiscriminatorValue que indica que Student engloba els Users amb role='S'.

Algunes de les etiquetes més útils són les que representen les relacions, ja sigui d'u a molts (@OneToMany) o de molts a molts (@ManyToMany). En aquest exemple s'especifica que un estudiant pot tenir molts tests, i que en esborrar un estudiant s'eliminaran els seus tests en cascada. A la següent línia s'indica que un estudiant pot estar matriculat a vàries assignatures, i que una assignatura pot tenir varis estudiants.

EntityManager i Spring

EntityManager és la interfície que permet interactuar amb el context de persistència. Un context de persistència és un conjunt d'instàncies d'entitats on a cada entitat de persistència li correspon una única instància. El EntityManager s'encarrega de controlar les instàncies de les entitats i els seus cicles de vida. La interfície d'EntityManager defineix els mètodes per interactuar amb el context de persistència, ja sigui per crear i eliminar entitats, buscar-ne o fer consultes sobre aquestes entitats.

```

<bean id="entityManagerFactory" class="org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean">
<property name="dataSource" ref="dataSource" />

<property name="loadTimeWeaver">
<bean class="org.springframework.instrument.classloading.ReflectiveLoadTimeWeaver"/> </property>

<property name="jpaVendorAdapter">
<bean class="org.springframework.orm.jpa.vendor.TopLinkJpaVendorAdapter">
<property name="showSql" value="false"/>
<property name="generateDdl" value="false"/>
<property name="databasePlatform" value="oracle.toplink.essentials.platform.database.PostgreSQLPlatform"/>
</bean>
</property>
</bean>

```

En aquest fragment de codi del fitxer applicationContext.xml s'hi indica el EntityManager, que s'utilitza TopLink i que gestor de base de dades és el PostgreSQL.

JPQL

Java Persistence Query Language (JPQL) és un llenguatge case sensitive de consulta orientat a objectes independent de plataformes que està definit com una part de l'especificació de JPA. JPQL està fortament inspirat en SQL, la seva sintaxis així ho demostra, però opera amb entitats JPA enlloc de fer-ho amb les taules de la base de dades, i això permet escriure consultes portables que funcionen independentment de la base de dades subjacent i també polimorfisme. Suporta consultes del tipus SELECT, UPDATE i DELETE.

A continuació es presenta un exemple de sentència JPQL:

```
getJpaTemplate().find("select st from Student st, Subject sj where st.Name=?2 and st.subjects=sj and sj.Id=?1",subjectId,name);
```

En aquest exemple s'obté l'estudiant amb nom name matriculat a l'assignatura subjectId. Per aconseguir-ho es fa un select però en el from s'indiquen les entitats, no les taules de la base de dades, i en el where es treballa amb els atributs de les entitats, no amb els camps de les taules com en SQL. Això proporciona independència entre la base de dades i la implementació de les classes que hi interaccionen, havent de modificar únicament, si varia alguna cosa en la base de dades, les etiquetes que es troben en l'entitat.

B.2 Patró DAO

El Patró DAO es basa en un raonament molt senzill però a la vegada molt útil. La idea és que la lògica de negoci sigui independent de la base de dades, és a dir, que no hi treballi directament sinó que ho faci amb una interfície que tingui els mètodes necessaris per a obtenir, modificar i eliminar la informació d'aquesta, amb la finalitat de que la interfície pugui disposar de varies implementacions i que els canvis a la tecnologia de persistència usada i fins i tot, a modificacions lleus al disseny de la base de dades siguin transparents a l'aplicació.

InnovaCampus disposa d'una interfície DAO per a cada taula de la base de dades, i seguint el citat raonament, la capa de negoci accedeix a les dades únicament a través d'aquestes interfícies.

StudentDAO.java: StudentDAO és la interfície per a interaccionar amb les dades d'estudiants de la base de dades.

```

public interface StudentDAO {

    public void addStudent(Student s);

    public Student findStudentById(String studentId);

    public List findStudentByName(String subjectId, String name);

    public List findStudentsBySubject(String subjectId);

    public void updateStudent(Student s);

    public void deleteStudent(String studentId);

}

```

B.3 Migració de tecnologies

Si en algun moment donat es decideix migrar de tecnologies, ja sigui canviar de gestor de base de dades, de ORM o utilitzar una altra classe que implementi una interfície DAO, Spring ens ho posa molt fàcil.

Per exemple, per indicar que la implementació de la interfície AnnouncementDAO es troba a la classe JPAAnnouncementDAO només cal afegir les següents sentències add/spring/applicationContext.xml:

```

<bean id="jpaAnnouncementDao" class="org.innovacampus.dao.jpa.JPAAnnouncementDAO">
<property name="entityManagerFactory" ref="entityManagerFactory" />
</bean>
<bean id="administrationManagement" class="org.innovacampus.services.impl.AdministrationManagementImpl">
[...]
<property name="announcementDAO" ref="jpaAnnouncementDao" />
[...]
</bean>

```

I si decidíssim utilitzar una implementació en Hibernate que es troba a HIBERNATEAnnouncementDAO doncs hauríem d'incloure la llibreria d'Hibernate al directori lib/libTomcat i actualitzar el fitxer dd/spring/applicationContext.xml (vegeu 3.4.2) especificant la ruta de la nova classe i associant-li a la interfície announcementDAO com es mostra a continuació.

```

<bean id="hbAnnouncementDao" class="org.innovacampus.dao.hibernate.HIBERNATEAnnouncementDAO">
<property name="entityManagerFactory" ref="entityManagerFactory" />
</bean>
<bean id="administrationManagement" class="org.innovacampus.services.impl.AdministrationManagementImpl">
[...]
<property name="announcementDAO" ref="hbAnnouncementDao" />
[...]
</bean>

```

Si el que pretenem és canviar el gestor de base de dades, per exemple per MySQL, hauríem d'especificar-ho al fitxer properties/jdbc.properties, indicant la url de MySQL, on es troba el driver, l'usuari i la contrasenya. També hauríem de referenciar la llibreria de MySQL i modificar el fitxer

dd/spring/applicationContext.xml configurant el EntityManager per a que treballi amb el gestor de la base de dades MySQL.

```
<bean id="dataSource" class="org.springframework.jdbc.datasource.DriverManagerDataSource">
[...]
```

```
</bean>
<bean id="entityManagerFactory" class="org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean">
[...]
```

```
<bean class="org.springframework.orm.jpa.vendor.TopLinkJpaVendorAdapter">
[...]
```

```
<property name="databasePlatform" value="rutaDeLaPlataformaMySQL"/>
</bean>
[...]
```

Canviar el gestor de base de dades implica haver de canviar també la implementació de les classes DAO, i per tant en el fitxer dd/spring/applicationContext.xml ho hem d'indicar.

```
<bean id="msAnnouncementDao" class="org.innovacampus.dao.mysql.MYSQLAnnouncementDAO">
<property name="entityManagerFactory" ref="entityManagerFactory" />
</bean>
<bean id="administrationManagement" class="org.innovacampus.services.impl.AdministrationManagementImpl">
[...]
```

```
<property name="announcementDAO" ref="msAnnouncementDao" />
[...]
```

```
</bean>
```

B.4 Transaccions

A l'hora d'implementar InnoVaCampus s'ha tingut en compte les transaccions a nivell d'accés a la base de dades, mitjançant el `getTransaction().begin` i `getTransaction().commit` tal i com es mostra en l'exemple, però no les transaccions a nivell de negoci.

JPAStudentDAO.java: JPAStudentDAO és una implementació en JPA de la interfície StudentDAO.

```
@Transactional
public class JPAStudentDAO extends JpaDaoSupport implements StudentDAO{

    public void addStudent(Student s) {
        getJpaTemplate().persist(s);
    }

    public void deleteStudent(String studentId) {
        Student s = findStudentById(studentId);
        if(s!=null){
            EntityManager em = getJpaTemplate().getEntityManagerFactory().createEntityManager();
            em.getTransaction().begin();
            em.createNativeQuery("DELETE FROM DIFFICULTY WHERE NIF='"+studentId+"'").executeUpdate(); em.getTransaction().commit();
            getJpaTemplate().remove(s);
        }
    }

    [...]

    public List findStudentByName(String subjectId, String name) {
        return getJpaTemplate().find("select st from Student st, Subject sj where st.Name=?2 and st.subjects=sj and sj.Id=?
1",subjectId,name);
    }

    [...]
}
```

En aquest fitxer hi apareixen operacions predefinides de JPA com `getJpaTemplate().persist(s)` que emmagatzema directament una entitat, o `em.getTransaction().begin()` i `em.getTransaction().commit()` que indiquen principi i fi d'una transacció.

En la darrera línia hi ha una consulta en JPQL, que com es pot comprovar, no és SQL.

Podeu trobar més informació sobre la capa de persistència a [4].

Apèndix C

La capa negoci

La capa de negoci és la que implementa la 'lògica de negoci' de l'aplicació. Aquesta lògica de negoci correspon essencialment a les funcionalitats que l'aplicació ofereix als seus clients. Aquesta capa rep les sol·licituds de la interfície per a retornar-li posteriorment els resultats, i es comunica amb la capa de persistència per interactuar amb la informació emmagatzemada.

Tota aquesta lògica ha estat implementada en un conjunt de classes aplegades dins el paquet `services`, ja que engloben tota la funcionalitat i serveis que dona l'aplicació. Dins d'aquest paquet, les funcionalitats estan agrupades temàticament; donant lloc a 4 subsistemes, cadascun s'encarregat d'una tasca diferent: la part més administrativa de l'aplicació (`AdministrationManagement`), la part encarregada de les assignatures (`SubjectManagement`), la part encarregada dels tests (`TestManagement`) i la part encarregada de les estadístiques (`StatisticsManagement`).

Els subsistemes implementen totes les funcionalitats que ha de complir l'aplicació, controlant totes les excepcions que poden donar-se en cada mètode i utilitzant les interfícies de les classes DAO per obtenir, modificar i eliminar la informació de la base de dades. Per poder interactuar correctament amb les classes DAO i amb la interfície s'utilitzen les classes del domini. Aquestes classes representen els conceptes que apareixen al domini de l'aplicació.

Cada subsistema disposa d'una interfície per estructurar una arquitectura de tres capes correcta, ja que això permet que si en un moment donat es decideix canviar-ne la implementació, la capa d'interfície no se'n ressentiria perquè interactua amb la interfície de la capa de negoci.

Tot seguit veurem quines classes formen part del domini i de què s'encarrega cada subsistema.

C.1 Classes del domini

Com hem comentat anteriorment, les classes del domini representen els usuaris que interaccionen amb l'aplicació i els objectes amb els quals interaccionen.

Moltes vegades les classes del domini contenen informació que s'ha de guardar i per tant al mateix temps que són classes de domini també són entitats que han de persistir en una base de dades. Però no sempre és així, ja que podríem tenir objectes amb els quals interactua l'usuari però que no ens interessa guardar la seva informació, com per exemple quan es fa una compra online, l'objecte del carret de la compra és un objecte que mentre anem fent la compra l'usuari interacciona amb ell (afegint o traient

productes) però que un cop l'usuari ha fet la comanda ja no el necessitem i l'objecte en sí del carret de la compra no cal guardar-ho en una base de dades. O, per exemple, objectes o coses que sempre seran de la mateixa forma, mai canviaran. Aquests objectes no cal guardar-los en una base de dades, perquè sempre sabrem com seran.

Llistat de classes que formen part del domini:

- Administrator
- Announcement
- Answer
- Objective
- Question
- Resource
- Result
- Student
- Subject
- SubjectAnnouncement
- Teacher
- Test
- Topic
- User

C.2 Els subsistemes

Durant l'anàlisi funcional d'InnovaCampus es va decidir estructurar la implementació dels serveis en funció de la seva funcionalitat.

De la mateixa manera que a moltes aplicacions web, sempre hi ha una part més administrativa, que en el nostre cas s'encarregaria dels usuaris, anuncis i la creació d'assignatures, aquesta seria una part. Una altra part fa referència a un element important dins d'InnovaCampus, que són les assignatures. I les altres dues parts corresponen als dos mòduls que hi havia fins ara, el de la generació de tests i el de les mostrar les estadístiques.

Tenint en compte tot el que s'ha dit, els subsistemes resultants són:

- AdministrationManagement: aquest subsistema s'encarrega de la part administrativa de l'aplicació. Gestiona els usuaris (tant professors com alumnes), donar-los d'alta, obtenir i modificar les seves dades i autenticar-los quan volen entrar a l'aplicació. Controla la creació d'anuncis, i obtenir i modificar la seva informació. Resol la creació d'assignatures, així com matricular alumnes i assignar professors a una assignatura.

- **SubjectManagement**: aquest subsistema es centra més en l'assignatura. S'encarrega d'afegir, modificar i esborrar temes i objectius. Una assignatura està dividida en temes i a la vegada els temes estan dividits en objectius.
- **TestManagement**: aquest subsistema s'ocupa de la part dels tests. És el responsable de la gestió de preguntes i respostes. Les preguntes formen part d'un objectiu en concret i cada pregunta té múltiples respostes. També s'encarrega d'associar un recurs a una pregunta i de la generació de tests.
- **StatisticsManagement**: aquest subsistema s'encarrega de la part d'estadístiques, d'obtenir tota la informació necessària per tal que el professor pugui visualitzar els resultats dels tests de forma gràfica, obtenir el resultat d'un test, saber quants estudiants han realitzat tests d'un objectiu o tema, saber quants han aprovat o suspès, mirar els resultats d'un període de temps determinat.

Cada subsistema disposa d'una interfície i la seva corresponent implementació, d'aquesta manera es redueix l'acoblament entre les capes. Seguidament mostrarem un fragment de la interfície i la implementació del subsistema **AdministrationManagement**.

Podeu trobar més informació sobre la capa negoci a [4].

Apèndix D

Casos d'ús

El diagrama de casos d'ús representa la forma com un client (Actor) opera amb el sistema, a més a més de la forma, tipus i ordre en què els elements interactuen.

Actor és un rol en què l'usuari interactua amb el sistema, no necessàriament representa a una persona en particular, sinó més aviat la tasca que realitza en front el sistema.

A continuació detallarem els casos d'ús classificats segons el rol d'usuari, la necessitat de tenir-los i el seus propòsits.

D.1 Casos d'ús comuns a tots els usuaris

Cas d'ús: Canviar el password d'usuari

Amb aquest cas d'ús, permetem que els diferents usuaris de l'aplicació, tinguin l'oportunitat de canviar el seu password, oferint així una eina de seguretat. El cas d'ús es desenvolupa de la mateixa manera que es faria en una típica aplicació on es canvia el password, és a dir, primer es demanaria el password antic i després el nou, i un cop validada la informació, el nou password quedaria registrat.

USUARI	SISTEMA
1.L'usuari es connecta a la pàgina i entra el seu login i password.	2. El sistema registra la informació.
3.L'usuari indica que vol canviar el seu password.	4. El sistema registra la situació i demana el password antic i el nou.
5.L'usuari introdueix el password actual i tot seguit el nou password.	6. El sistema registra la situació i procedeix a fer la modificació.
7. L'usuari veu que tot ha anat correctament.	

Casos d'error:

- 1. El login i el password no són correctes. S'ha denegat l'accés a la pàgina i s'informa de l'error.
- Password no vàlid. No s'efectua la modificació i s'informa de l'error.
- 5. Password antic incorrecte. No s'efectua el canvi i s'informa de l'error.

Cas d'ús: Recordar clau

Aquest cas d'ús detalla els passos que han de seguir professors , alumnes i administrador per a recordar la seva clau d'accés al sistema. L'únic q faran és introduir el seu DNI dins d'un formulari i tot seguit rebran un correu electrònic amb la clau. No es tracta ben bé d'un recordatori de clau, ja que degut a l'encriptació de la mateix, el que es farà és generar una nova clau aleatòriament i s'enviarà a l'usuari. Aquesta clau podrà ser canviada quan es desitgi (Veure cas d'ús modificar Password).

USUARI	SISTEMA
1.L'usuari indica que vol recordar el Password	2. El sistema registra la informació i demana a l'usuari la introducció del seu DNI.
3. El professor o alumne introdueix el seu DNI	4. El sistema registra la situació, procedeix a generar una nova clau aleatòriament , l'encrypta i l'envia al correu de l'usuari.
5.El professor o alumne rep el correu amb la clau i veu que tot ha anat correctament.	

Casos d'error:

- 3. El DNI no és correcte.

D.2 Casos d'ús específics de professors i alumnes

Cas d'ús: Modificar les dades d'usuari

Aquest cas d'ús detalla els passos que ha de seguir l'usuari de l'aplicació, per a poder modificar les seves dades. L'usuari introduirà les seves dades i l'aplicació s'encarregarà de validar-les i posteriorment registrar els canvis. Una vegada validats els canvis quedaran emmagatzemats.

USUARI	SISTEMA
1.L'usuari es connecta a la pàgina i entra el seu login i password.	2. El sistema registra la informació.
3.L'usuari indica que vol modificar les seves dades.	4. El sistema registra la situació i mostra les dades actuals per tal que l'usuari, si ho desitja, les modifiqui.
5.L'usuari introdueix les dades que desitja canviar.	6. El sistema registra la informació i procedeix a la modificació.
7.L'usuari veu que tot ha anat correctament.	

Casos d'error:

- 1. El login i el password no són correctes. S'ha denegat l'accés a la pàgina i s'informa de l'error.
- 5. Alguna de les dades introduïdes pot ser incorrecta. El professor queda amb les dades anteriors i s'informa de l'error.

Cas d'ús: Veure anuncis de l'administrador

Aquest cas d'ús detalla els passos que han de seguir professors i alumnes per a veure els anuncis que introdueix l'administrador. Aquests anuncis podran estar destinats a professors, alumnes o a ambdós. Es visualitzen de la mateixa manera. El professor o alumne accedeix al sistema, i dins de la seva pàgina inici veurà una taula amb anuncis de l'administrador, dins de la taula hi ha l'opció veure tots els missatges que mostra tots els anuncis introduïts per l'administrador.

USUARI	SISTEMA
1.El professor o alumne indica que vol registrar-se al sistema.	2. El sistema registra la informació i demana a l'usuari la introducció del DNI, correu i la clau d'accés.
3. El professor o alumne introdueix les dades (DNI, correu i clau dues vegades).	4. El sistema registra la situació, procedeix a l'alta i envia un correu electrònic amb les dades a l'usuari.
5.El professor o alumne veu que tot ha anat correctament.	

Casos d'error:

- 1. El login i el password no són correctes. S'ha denegat l'accés a la pàgina i s'informa de l'error.

Cas d'ús: Veure anuncis del taulell d'anuncis d'una assignatura

Aquest cas d'ús detalla els passos que han de seguir professors i alumnes per a veure els anuncis del taulell d'anuncis d'una assignatura. L'alumne o professor es connectarà al sistema i accedirà al camp de treball de l'assignatura, llavors podrà accedir al taulell d'anuncis de la mateixa per visualitzar els missatges.

USUARI	SISTEMA
1.El professor o alumne es connecta a la pàgina i entra el seu login i password.	2. El sistema registra la informació.
3.El professor o alumne indica que vol visualitzar els missatges introduïts per l'administrador.	4. El sistema registra la situació i genera una llista amb tots els missatges de l'administrador.
5.El professor o alumne veu que tot ha anat correctament.	

Casos d'error:

- 1. El login i el password no són correctes. S'ha denegat l'accés a la pàgina i s'informa de l'error.

Cas d'ús: Veure anuncis del taulell d'anuncis d'una assignatura

Aquest cas d'ús detalla els passos que han de seguir professors i alumnes per a veure els anuncis del taulell d'anuncis d'una assignatura. L'alumne o professor es connectarà al sistema i accedirà al camp de treball de l'assignatura, llavors podrà accedir al taulell d'anuncis de la mateixa per visualitzar els missatges.

USUARI	SISTEMA
1.El professor o alumne es connecta a la pàgina i entra el seu login i password.	2. El sistema registra la informació.
3.El professor o alumne accedeix al camp de l'assignatura.	4. El sistema registra la situació.
5.El professor o alumne indica que vol visualitzar els anuncis del taulell d'anuncis.	6. El sistema registra la situació i procedeix a generar una llista de tots els anuncis del taulell.
7. El professor o alumne veu que tot ha anat correctament.	

Casos d'error:

- 1. El login i el password no són correctes. S'ha denegat l'accés a la pàgina i s'informa de l'error.

D.3 Casos d'ús específics de l'administrador

Cas d'ús: Donar d'alta un professor

Aquest cas d'ús detalla els passos que ha de seguir l'administrador per a donar d'alta un nou professor al sistema. L'administrador serà l'encarregat d'introduir les dades del professor, que quedarà emmagatzemat amb el seu corresponent DNI, nom i cognoms, correu i telèfon.

USUARI	SISTEMA
1.L'administrador es connecta a la pàgina i entra el seu login i password.	2. El sistema registra la informació.
3.L'administrador indica que vol donar d'alta un nou professor.	4. El sistema registra la situació i demana les dades del nou professor.
5.L'administrador introdueix les dades del professor tals com: Nom, cognoms, DNI, telèfon, correu electrònic.	6. El sistema registra la informació i procedeix a l'alta del professor.
7.L'administrador veu que tot ha anat correctament.	

Casos d'error:

- 1. El login i el password no són correctes. S'ha denegat l'accés a la pàgina i s'informa de l'error.
- 5. Alguna de les dades introduïdes pot ser incorrecta. El professor nou no queda registrat i s'informa de l'error. Casos possibles: DNI invàlid, direcció correu electrònic incorrecta.

Cas d'ús: Donar de baixa un professor

Aquest cas d'ús detalla els passos que ha de seguir l'administrador per a donar de baixa un professor. L'administrador serà l'encarregat d'introduir el DNI del professor a donar de baixa.

USUARI	SISTEMA
1.L'administrador es connecta a la pàgina i entra el seu login i password.	2. El sistema registra la informació.
3.L'administrador indica que vol donar de baixa un professor.	4. El sistema registra la situació i genera una llista dels professors.
5.L'administrador selecciona de la llista el professor a donar de baixa.	6. El sistema registra la informació i procedeix a la baixa del professor.
7.L'administrador veu que tot ha anat correctament.	

Casos d'error:

- 1. El login i el password no són correctes. S'ha denegat l'accés a la pàgina i s'informa de l'error.

Cas d'ús: Donar d'alta un anunci

Aquest cas d'ús detalla els passos que ha de seguir l'administrador per a donar d'alta un nou anunci. L'administrador indicarà que vol donar d'alta un nou anunci, elegirà el destinatari i procedirà a escriure el contingut del mateix. Un cop escrit l'enviarà i es produirà l'alta.

USUARI	SISTEMA
1.L'administrador es connecta a la pàgina i entra el seu login i password.	2. El sistema registra la informació.
3.L'administrador indica que vol donar de baixa un anunci.	4. El sistema registra la situació i demana a l'administrador que elegeixi el destinatari de l'anunci que vol eliminar.
5.L'administrador elegeix el destinatari de l'anunci (professors, alumnes, ambdós, anunci de caràcter general) per tal de veure els anuncis que té aquell destinatari.	6. El sistema registra la informació i genera una llista amb els anuncis d'aquell destinatari.
7.L'administració selecciona l'anunci que desitja esborrar.	8. El sistema registra la informació i esborra l'anunci seleccionat.
9. L'administrador veu que tot ha anat correctament.	

Casos d'error:

- 1. El login i el password no són correctes. S'ha denegat l'accés a la pàgina i s'informa de l'error.

Cas d'ús: Donar de baixa un anunci

Aquest cas d'ús detalla els passos que ha de seguir l'administrador per a donar de baixa un anunci. L'administrador indicarà que vol donar de baixa un anunci, seleccionarà el destinatari de l'anunci i visualitzarà la llista d'anuncis d'aquell destinatari, només li caldrà elegir l'anunci i es procedirà a la baixa corresponent.

USUARI	SISTEMA
1.L'administrador es connecta a la pàgina i entra el seu login i password.	2. El sistema registra la informació.
3.L'administrador indica que vol donar de baixa un anunci.	4. El sistema registra la situació i demana a l'administrador que elegeixi el destinatari de l'anunci que vol eliminar.
5.L'administrador elegeix el destinatari de l'anunci (professors, alumnes, ambdós, anunci de caràcter general) per tal de veure els anuncis que té aquell destinatari.	6. El sistema registra la informació i genera una llista amb els anuncis d'aquell destinatari.
7.L'administració selecciona l'anunci que desitja esborrar.	8. El sistema registra la informació i esborra l'anunci seleccionat.
9. L'administrador veu que tot ha anat correctament.	

Casos d'error:

- 1. El login i el password no són correctes. S'ha denegat l'accés a la pàgina i s'informa de l'error.

D.4 Casos d'ús específics dels professors

Cas d'ús: Donar d'alta una assignatura

Aquest cas d'ús detalla els passos que ha de seguir un professor per a donar d'alta una nova assignatura. Al professor se li delegarà la responsabilitat d'introduir les dades de l'assignatura, temes, objectius, preguntes, respostes, així com la possibilitat de matricular estudiants. Es validaran les dades abans d'actualitzar els canvis.

USUARI	SISTEMA
1.L'administrador es connecta a la pàgina i entra el seu login i password.	2. El sistema registra la informació.
3.L'administrador indica que vol donar de baixa un professor.	4. El sistema registra la situació i genera una llista dels professors.
5.L'administrador selecciona de la llista el professor a donar de baixa.	6. El sistema registra la informació i procedeix a la baixa del professor.
7.L'administrador veu que tot ha anat correctament.	

Casos d'error:

- 1. El login i el password no són correctes. S'ha denegat l'accés a la pàgina i s'informa de l'error.
- 5. El codi de l'assignatura ja existeix o bé no es vàlid. No es dona d'alta l'assignatura i s'informa de l'error.

Cas d'ús: Donar de baixa una assignatura

Aquest cas d'ús detalla els passos que ha de seguir un professor per a donar de baixa una assignatura. El professor seleccionarà una de les assignatures per tal de procedir a l'eliminació de l'assignatura i tota la informació associada a aquesta. Cal dir que quan s'elimina una assignatura l'aplicació esborrarà tots els temes que tingui associats, així com els objectius, preguntes i respostes pertinents. També eliminarà els registres de la taula matriculats que coincideixin amb els codis d'alumnes que tingui aquesta assignatura, aquest mateix procés es repetirà amb els resultats. Per tant l'eliminació també implica una ajustament de les estadístiques corresponents.

USUARI	SISTEMA
1.El professor es connecta a la pàgina i entra el seu login i password.	2. El sistema registra la informació.
3.El professor indica que vol donar de baixa una assignatura.	4. El sistema registra la situació i genera una llista de les assignatures del professor.
5.El professor escull una de les assignatures de la llista.	6. El sistema registra la informació i procedeix a la baixa de l'assignatura. Com a conseqüència també s'esborraran tots els alumnes matriculats, professors responsables i tota la informació de l'assignatura.
7.El professor veu que tot ha anat correctament.	

Casos d'error:

- 1. El login i el password no són correctes. S'ha denegat l'accés a la pàgina i s'informa de l'error.

Cas d'ús: Modificar dades d' una assignatura

Aquest cas d'ús detalla els passos que ha de seguir un professor per a modificar les dades d'una assignatura. El professor seleccionarà l'assignatura que desitja modificar i s'encarregarà d'introduir les dades a modificar, les quals l'aplicació validarà abans de ser registrades. Un cop validades quedaran emmagatzemades.

USUARI	SISTEMA
1.El professor es connecta a la pàgina i entra el seu login i password.	2. El sistema registra la informació.
3.El professor indica que vol modificar les dades d'una assignatura.	4. El sistema registra la situació i genera una llista de les assignatures del professor.
5.El professor escull una de les assignatures de la llista.	6. El sistema registra la informació i mostra la informació actual de l'assignatura per tal que el professor faci els canvis corresponents.
7.El professor introdueix les dades que desitja canviar.	8. El sistema registra la informació i procedeix a la modificació de les dades.
9. El professor veu que tot ha anat correctament.	

Casos d'error:

- 1. El login i el password no són correctes. S'ha denegat l'accés a la pàgina i s'informa de l'error.

- 7. Alguna de les dades introduïdes no és correcta. S'informa de l'error.

Cas d'ús: Reiniciar una assignatura

En aquest cas d'ús, el professor podrà, donada una assignatura, reiniciar-la. El procés de reiniciar una assignatura consistirà en desmatricular-ne a tots els estudiants, així com totes les dades de l'assignatura en concret, és a dir, s'esborraran les referències a cada alumne per a cada un dels testos, i a la vegada totes les referències a l'entitat resultats que també seran eliminats.

USUARI	SISTEMA
1.El professor es connecta a la pàgina i entra el seu login i password.	2. El sistema registra la informació.
3.El professor indica que reiniciar una assignatura.	4. El sistema registra la situació i genera una llista de les assignatures del professor.
5.El professor escull una de les assignatures de la llista.	6. El sistema registra la informació i procedeix a reiniciar l'assignatura. Com a conseqüència també s'esborraran tots els alumnes matriculats i les dades a les quals faci referència en les entitats testos i resultats.
7.El professor veu que tot ha anat correctament.	

Casos d'error:

- 1. El login i el password no són correctes. S'ha denegat l'accés a la pàgina i s'informa de l'error.

Cas d'ús: Donar d'alta un professor a una assignatura

Aquest cas d'ús detalla els passos que ha de seguir un professor per a donar d'alta un altre professor a l'assignatura. El professor seleccionarà d'una llista el professor a qui delegarà l'assignatura. Per a poder donar d'alta un professor a una assignatura, aquest professor ha d'haver estat donat d'alta a l'aplicació prèviament.

USUARI	SISTEMA
1.El professor es connecta a la pàgina i entra el seu login i password.	2. El sistema registra la informació.
3.El professor accedeix a l'espai de treball de l'assignatura a la que vol afegir un responsable.	4. El sistema registra la situació.
5.El professor indica que vol afegir un nou professor a l'assignatura.	6. El sistema registra la informació i genera una llista dels professors candidats.
7.El professor escull un dels professors de la llista.	8. El sistema registra la informació i procedeix a l'alta del professor escollit a l'assignatura.
9. El professor veu que tot ha anat correctament.	

Casos d'error:

- 1. El login i el password no són correctes. S'ha denegat l'accés a la pàgina i s'informa de l'error.

Cas d'ús: Donar de baixa un professor de l' assignatura

Aquest cas d'ús detalla els passos que ha de seguir un professor per a donar de baixa un professor de l'assignatura. El professor seleccionarà d'una llista el professor a qui vulgui treure la responsabilitat, i un cop seleccionat i validat, els canvis quedaran emmagatzemats i tal professor no formarà part dels responsables de l'assignatura.

USUARI	SISTEMA
1.El professor es connecta a la pàgina i entra el seu login i password.	2. El sistema registra la informació.
3.El professor accedeix a l'espai de treball de l'assignatura.	4. El sistema registra la situació.
5.El professor indica que vol eliminar un professor de l'assignatura.	6. El sistema registra la informació i genera una llista dels professors candidats.
7.El professor escull un dels professors de la llista.	8. El sistema registra la informació i procedeix a la baixa del professor escollit a l'assignatura.
9. El professor veu que tot ha anat correctament.	

Casos d'error:

- 1. El login i el password no són correctes. S'ha denegat l'accés a la pàgina i s'informa de l'error.

Cas d'ús: Donar d'alta un alumne a l'assignatura

Aquest cas d'ús detalla els passos que ha de seguir un professor per a donar d'alta un nou alumne a l'assignatura. El professor introduirà les dades de l'alumne, i un cop validades l'aplicació efectuarà l'alta corresponent.

USUARI	SISTEMA
1.El professor es connecta a la pàgina i entra el seu login i password.	2. El sistema registra la informació.
3.El professor accedeix a l'espai de treball de l'assignatura a la que vol matricular un alumne.	4. El sistema registra la situació.
5.El professor indica que vol matricular un nou alumne a l'assignatura.	6. El sistema registra la situació i demana al professor les dades del nou alumne.
7.El professor introdueix les dades de l'alumne tals com el nom, cognoms i el dni.	8. El sistema registra la informació i procedeix a la matriculació de l'alumne a l'assignatura.
9. El professor veu que tot ha anat correctament.	

Casos d'error:

- 1. El login i el password no són correctes. S'ha denegat l'accés a la pàgina i s'informa de l'error.
- 7. El DNI introduït pel professor no és vàlid. No es dona d'alta a l'alumne i s'informa de l'error.

Cas d'ús: Donar d'alta una llista d'alumnes a l'assignatura mitjançant un fitxer

Aquest cas d'ús detalla els passos que ha de seguir un professor per a donar d'alta a l'assignatura una llista d'alumnes emmagatzemada en un fitxer. El professor introduirà el fitxer i l'aplicació s'encarregarà de la matriculació.

USUARI	SISTEMA
1.El professor es connecta a la pàgina i entra el seu login i password.	2. El sistema registra la informació.
3.El professor accedeix a l'espai de treball de l'assignatura a la que vol fer la matriculació.	4. El sistema registra la situació.
5.El professor indica que vol dur a terme una matriculació d'alumnes mitjançant un fitxer.	6. El sistema registra la situació i demana al professor el fitxer.
7.El professor selecciona el fitxer.	8. El sistema registra la informació i procedeix a la matriculació de l'alumne a l'assignatura.
9. El professor veu que tot ha anat correctament.	

Casos d'error:

- 1. El login i el password no són correctes. S'ha denegat l'accés a la pàgina i s'informa de l'error.
- 7. La ruta del fitxer introduïda no és correcta o el fitxer no existeix. S'informa de l'error.
- 7. El format del fitxer no és el correcte. S'informa de l'error.

Cas d'ús: Donar de baixa un alumne de l' assignatura

Aquest cas d'ús detalla els passos que ha de seguir un professor per a donar de baixa un alumne de l'assignatura. El professor selecciona l'alumne de la llista de matriculats i l'aplicació procedirà a desmatricular-lo de l'assignatura. També s'esborraran els testos que hagi realitzat i totes les puntuacions que tingues emmagatzemades.

USUARI	SISTEMA
1.El professor es connecta a la pàgina i entra el seu login i password.	2. El sistema registra la informació.
3.El professor accedeix a l'espai de treball de l'assignatura a la que vol desmatricular un alumne.	4. El sistema registra la situació.
5.El professor indica que vol desmatricular un nou alumne a l'assignatura.	6. El sistema registra la situació i genera una llista dels alumnes de l'assignatura.
7.El professor selecciona l'alumne que vol donar de baixa.	8. El sistema registra la informació i procedeix a la desmatriculació de l'alumne de l'assignatura. També borra tots els testos realitzats per l'alumne en els quals hi ha una referència a l'assignatura, el mateix passa amb els resultats.
9. El professor veu que tot ha anat correctament.	

Casos d'error:

- 1. El login i el password no són correctes. S'ha denegat l'accés a la pàgina i s'informa de l'error.

Cas d'ús: Donar d'alta un tema a l'assignatura

Aquest cas d'ús detalla els passos que ha de fer un professor per tal de donar d'alta un tema a l'assignatura. El professor introduirà les dades del tema. Al finalitzar el cas d'ús el tema haurà quedat emmagatzemat i quedarà lligat a l'assignatura. El tema ja estarà preparat per a poder afegir-hi objectius.

USUARI	SISTEMA
1.El professor es connecta a la pàgina i entra el seu login i password.	2. El sistema registra la informació.
3.El professor accedeix a l'espai de treball de l'assignatura a la que vol afegir el tema.	4. El sistema registra la situació.
5.El professor indica que vol afegir un nou tema.	6. El sistema registra la situació i demana a l'usuari el títol del tema i la seva descripció.
7. El professor introdueix les dades del tema (nom i descripció).	8. El sistema registra la informació i procedeix a l'alta del tema.
9. El professor veu que tot ha anat correctament.	

Casos d'error:

- 1. El login i el password no són correctes. S'ha denegat l'accés a la pàgina i s'informa de l'error.
- 7. Les dades introduïdes no són correctes. No es produeix la inserció del tema i s'informa de l'error.

Cas d'ús: Donar de baixa un tema de l'assignatura

Aquest cas d'ús detalla els passos que ha de fer un professor per tal de donar de baixa un tema de l'assignatura. El professor seleccionarà el tema que vol donar de baixa i l'aplicació s'encarregarà d'eliminar-lo. S'ha de tenir en compte que al eliminar un tema, també s'eliminaran les dades que coincideixin amb la referència a aquest com són els objectius, les preguntes, respostes, resultats i per tant els testos.

USUARI	SISTEMA
1.El professor es connecta a la pàgina i entra el seu login i password.	2. El sistema registra la informació.
3.El professor accedeix a l'espai de treball de l'assignatura de la qual vol eliminar el tema.	4. El sistema registra la situació.
5.El professor indica que vol eliminar un tema.	6. El sistema registra la situació i genera una llista dels temes de l'assignatura.
7. El professor selecciona el tema de la llista .	8.El sistema registra la informació i procedeix a la baixa del tema, i com a conseqüència també s'eliminaran totes les dades relacionades.
9. El professor veu que tot ha anat correctament.	

Casos d'error:

- 1. El login i el password no són correctes. S'ha denegat l'accés a la pàgina i s'informa de l'error.

Cas d'ús: Modificar un tema de l'assignatura

Aquest cas d'ús detalla els passos que ha de fer un professor per tal de modificar un tema. El professor seleccionarà el tema i s'encarregarà de modificar-ne les dades. El sistema modificarà les dades introduïdes pel professor abans de ser registrades. Un cop finalitzat el cas d'ús el tema quedarà emmagatzemat amb la nova informació.

USUARI	SISTEMA
1.El professor es connecta a la pàgina i entra el seu login i password.	2. El sistema registra la informació.
3.El professor accedeix a l'espai de treball de l'assignatura a la que vol modificar el tema.	4. El sistema registra la situació.
5.El professor indica que vol modificar un tema.	6. El sistema registra la situació i genera una llista dels temes de l'assignatura.
7. El professor selecciona el tema de la llista.	8. El sistema registra la situació i mostra la informació actual del tema per tal que el professor la pugui modificar.
9. El professor introdueix les dades que desitja canviar.	10. El sistema registra les noves dades i queden emmagatzemades.
11. El professor veu que tot ha anat correctament.	

Casos d'error:

- 1. El login i el password no són correctes. S'ha denegat l'accés a la pàgina i s'informa de l'error.
- 9. Les dades introduïdes no són correctes. No es produeix la modificació del tema i s'informa de l'error.

Cas d'ús: Donar d'alta un objectiu a un tema de l'assignatura

Aquest cas d'ús detalla els passos que ha de fer un professor per tal de donar d'alta un objectiu dins d'un tema de l'assignatura. El professor seleccionarà el tema on vol afegir l'objectiu i introduirà les dades de l'objectiu. Al finalitzar el cas d'ús l'objectiu haurà quedat emmagatzemat i quedarà lligat al tema de l'assignatura. L'objectiu ja estarà preparat per a poder afegir-hi preguntes i respostes.

USUARI	SISTEMA
1.El professor es connecta a la pàgina i entra el seu login i password.	2. El sistema registra la informació.
3.El professor accedeix a l'espai de treball de l'assignatura de la qual vol eliminar l'objectiu	4. El sistema registra la situació.
5.El professor indica que vol eliminar un objectiu.	6. El sistema registra la situació i genera una llista dels temes de l'assignatura.
7.El professor selecciona el tema del qual vol eliminar l'objectiu.	8. El sistema registra la informació i genera la llista d'objectius del tema.
9. El professor selecciona l'objectiu de la llista .	10.El sistema registra la informació i procedeix a la baixa de l'objectiu, i com a conseqüència també s'eliminaran totes les dades relacionades.
11. El professor veu que tot ha anat correctament.	

Casos d'error:

- 1. El login i el password no són correctes. S'ha denegat l'accés a la pàgina i s'informa de l'error.
- 9. Les dades introduïdes no són correctes. El nombre de preguntes no és un nombre natural. No es produeix la inserció de l'objectiu i s'informa de l'error.

Cas d'ús: Donar de baixa un objectiu d'un tema de l'assignatura

Aquest cas d'ús detalla els passos que ha de fer un professor per tal de donar de baixa un objectiu d'un tema de l'assignatura. El professor selecciona el tema d'on vol eliminar l'objectiu, tot seguit seleccionarà l'objectiu que vol donar de baixa. S'ha de tenir en compte que al eliminar un objectiu, també s'eliminaran les dades que coincideixin amb la referència a aquest com són les preguntes, respostes, resultats i per tant els testos.

USUARI	SISTEMA
1.El professor es connecta a la pàgina i entra el seu login i password.	2. El sistema registra la informació.
3.El professor accedeix a l'espai de treball de l'assignatura de la qual vol eliminar l'objectiu	4. El sistema registra la situació.
5.El professor indica que vol eliminar un objectiu.	6. El sistema registra la situació i genera una llista dels temes de l'assignatura.
7.El professor selecciona el tema del qual vol eliminar l'objectiu.	8. El sistema registra la informació i genera la llista d'objectius del tema.
9. El professor selecciona l'objectiu de la llista .	10.El sistema registra la informació i procedeix a la baixa de l'objectiu, i com a conseqüència també s'eliminaran totes les dades relacionades.
11. El professor veu que tot ha anat correctament.	

Casos d'error:

- 1. El login i el password no són correctes. S'ha denegat l'accés a la pàgina i s'informa de l'error.

Cas d'ús: Modificar un objectiu d'un tema de l'assignatura

Aquest cas d'ús detalla els passos que ha de fer un professor per tal de modificar un objectiu. El professor seleccionarà el tema que conté l'objectiu, i tot seguit seleccionarà l'objectiu i s'encarregarà de modificar-ne les dades. El sistema modificarà les dades introduïdes pel professor abans de ser registrades. Un cop finalitzat el cas d'ús l'objectiu quedarà emmagatzemat amb la nova informació.

USUARI	SISTEMA
1.El professor es connecta a la pàgina i entra el seu login i password.	2. El sistema registra la informació.
3.El professor accedeix a l'espai de treball de l'assignatura a la que vol modificar l'objectiu.	4. El sistema registra la situació.
5.El professor indica que vol modificar un objectiu.	6. El sistema registra la situació i genera una llista dels temes de l'assignatura.
7.El professor selecciona el tema en el qual vol modificar l'objectiu.	8. El sistema registra la informació i genera la llista d'objectius del tema.
9. El professor selecciona l'objectiu de la llista.	10. El sistema registra la situació i mostra les dades de l'objectiu seleccionat per tal de ser modificades.
11. El professor introdueix les dades que desitja canviar.	12. El sistema registra les noves dades i queden emmagatzemades.
13. El professor veu que tot ha anat correctament.	

Casos d'error:

- 1.El login i el password no són correctes. S'ha denegat l'accés a la pàgina i s'informa de l'error.
- 11.Les dades introduïdes no són correctes. El nombre de preguntes no és un nombre natural. No es produeix la modificació de l'objectiu i s'informa de l'error.

Cas d'ús: Donar d'alta una pregunta a un objectiu

Aquest cas d'ús detalla els passos que ha de fer un professor per tal de donar d'alta una pregunta dins d'un objectiu contingut en un dels temes de l'assignatura. El professor seleccionarà el tema i tot seguit l'objectiu en el qual vol donar d'alta una nova pregunta, ell serà l'encarregat d'introduir les dades de la pregunta, així com també, si ho desitja, podrà donar d'alta respostes associades a la pregunta generada.

USUARI	SISTEMA
1.El professor es connecta a la pàgina i entra el seu login i password.	2. El sistema registra la informació.
3.El professor accedeix a l'espai de treball de l'assignatura.	4. El sistema registra la situació.
5.El professor indica que vol afegir una nova pregunta.	6. El sistema registra la situació i genera una llista dels temes de l'assignatura.
7.El professor selecciona el tema en el qual es troba l'objectiu al qual vol afegir una nova pregunta.	8. El sistema registra la informació i genera una llista dels objectius del tema.
9. El professor selecciona l'objectiu en el qual vol afegir la pregunta.	10. El sistema registra la situació i demana el text i número de respostes de la pregunta.
11. El professor introdueix les dades de la pregunta, tals com, el text de la pregunta i el número de respostes.	12. El sistema registra la informació, dóna d'alta la pregunta i tot seguit demanarà que l'usuari introdueixi les respostes corresponents.
13. El professor introdueix les diferents respostes i selecciona la correcta.	14. El sistema registra la informació i dóna d'alta les respostes.
15. El professor veu que tot ha anat correctament.	

Casos d'error:

- 1. El login i el password no són correctes. S'ha denegat l'accés a la pàgina i s'informa de l'error.
- 11. Les dades introduïdes no són correctes. El nombre de respostes no és un nombre natural. No es produeix la inserció de la pregunta i s'informa de l'error.
- 13. Les dades introduïdes no són correctes. No es produeix la inserció de les respostes i s'informa de l'error.

Cas d'ús: Donar de baixa una pregunta d'un objectiu

Aquest cas d'ús detalla els passos que ha de fer un professor per tal de donar de baixa una pregunta. El professor seleccionarà el tema i tot seguit l'objectiu que conté la pregunta que es vol eliminar, seleccionarà la qüestió en concret i es procedirà a la seva eliminació. Aquest procés anirà lligat amb l'eliminació de les respostes pertinents, així com les referències que tinguin en les entitats resultats i testos.

USUARI	SISTEMA
1.El professor es connecta a la pàgina i entra el seu login i password.	2. El sistema registra la informació.
3.El professor accedeix a l'espai de treball de l'assignatura.	4. El sistema registra la situació.
5.El professor indica que vol eliminar una pregunta.	6. El sistema registra la situació i genera una llista dels temes de l'assignatura.
7.El professor selecciona el tema en el qual es troba l'objectiu que conté la pregunta.	8. El sistema registra la informació i genera una llista dels objectius del tema.
9. El professor selecciona l'objectiu.	10. El sistema registra la situació i genera una llista de les preguntes de l'objectiu.
11. El professor escull la pregunta que vol eliminar.	12. El sistema registra la informació, elimina la pregunta i com a conseqüència les respostes, resultats i textos d'aquesta pregunta.
13. El professor veu que tot ha anat correctament.	

Casos d'error:

- 1. El login i el password no són correctes. S'ha denegat l'accés a la pàgina i s'informa de l'error.

Cas d'ús: Modificar una pregunta

Aquest cas d'ús detalla els passos que ha de fer un professor per tal de modificar les dades d'una pregunta. El professor seleccionarà el tema, l'objectiu, així com la pregunta que vol modificar i introduirà les dades corresponents. El sistema validarà les dades introduïdes i tot seguit procedirà a la modificació corresponent.

USUARI	SISTEMA
1.El professor es connecta a la pàgina i entra el seu login i password.	2. El sistema registra la informació.
3.El professor accedeix a l'espai de treball de l'assignatura.	4. El sistema registra la situació.
5.El professor indica que vol modificar una pregunta.	6. El sistema registra la situació i genera una llista dels temes de l'assignatura.
7.El professor selecciona el tema en el qual es troba l'objectiu que conté la pregunta que vol modificar.	8. El sistema registra la informació i genera una llista dels objectius del tema.
9. El professor selecciona l'objectiu on es troba la pregunta.	10. El sistema registra la situació i genera una llista de les preguntes.
11. El professor selecciona la pregunta que vol modificar.	12. El sistema registra la situació i mostra el text de la pregunta per tal que el professor la modifiqui.
13. El professor introdueix les dades de la pregunta que desitja canviar.	14. El sistema registra la informació i procedeix a la modificació.
15. El professor veu que tot ha anat correctament.	

Casos d'error:

- 1. El login i el password no són correctes. S'ha denegat l'accés a la pàgina i s'informa de l'error.
- 13. Les dades introduïdes no són correctes. No es produeix la inserció de la pregunta i s'informa de l'error.

Cas d'ús: Donar d'alta una resposta en una pregunta

Aquest cas d'ús detalla els passos que ha de fer un professor per tal de donar d'alta una nova resposta dins d'una pregunta. El professor selecciona el tema, l'objectiu, així com la pregunta a la qual vol afegir una nova resposta, i tot seguit, introduirà les dades corresponents a la nova resposta. El sistema validarà les dades introduïdes i procedirà a la inserció corresponent.

USUARI	SISTEMA
1.El professor es connecta a la pàgina i entra el seu login i password.	2. El sistema registra la informació.
3.El professor accedeix a l'espai de treball de l'assignatura.	4. El sistema registra la situació.
5.El professor indica que vol afegir una resposta nova dins d'una pregunta existent .	6. El sistema registra la situació i genera una llista dels temes de l'assignatura.
7.El professor selecciona el tema en el qual es troba l'objectiu que conté la pregunta a la qual vol afegir una nova resposta.	8. El sistema registra la informació i genera una llista dels objectius del tema.
9. El professor selecciona l'objectiu on es troba la pregunta.	10. El sistema registra la situació i genera una llista amb les preguntes de l'objectiu.
11. El professor selecciona la pregunta.	12. El sistema registra la situació i demana les dades de la nova resposta.
13. El professor introdueix les dades de la nova resposta.	14. El sistema registra la informació i procedeix a la inserció.
15. El professor veu que tot ha anat correctament.	

Casos d'error:

- 1. El login i el password no són correctes. S'ha denegat l'accés a la pàgina i s'informa de l'error.
- 13. Les dades introduïdes no són correctes. No es produeix la inserció i s'informa de l'error.

Cas d'ús: Donar de baixa una resposta d'una pregunta

Aquest cas d'ús detalla els passos que ha de fer un professor per tal de donar de baixa una resposta. El professor seleccionarà el tema, l'objectiu, així com la pregunta i posteriorment la resposta que es vol eliminar.

USUARI	SISTEMA
1.El professor es connecta a la pàgina i entra el seu login i password.	2. El sistema registra la informació.
3.El professor accedeix a l'espai de treball de l'assignatura.	4. El sistema registra la situació.
5.El professor indica que vol eliminar una resposta existent.	6. El sistema registra la situació i genera una llista dels temes de l'assignatura.
7.El professor selecciona el tema en el qual es troba l'objectiu que conté la pregunta que a la vegada conté la resposta que es vol donar de baixa.	8. El sistema registra la informació i genera una llista dels objectius del tema.
9. El professor selecciona l'objectiu on es troba la pregunta.	10. El sistema registra la situació i genera una llista amb les preguntes de l'objectiu.
11. El professor selecciona la pregunta.	12. El sistema registra la situació i genera una llista de les respostes corresponents a la pregunta escollida.
13. El professor selecciona la resposta que es vol donar de baixa.	14. El sistema registra la informació i procedeix a l'eliminació.
15. El professor veu que tot ha anat correctament.	

Casos d'error:

- 1. El login i el password no són correctes. S'ha denegat l'accés a la pàgina i s'informa de l'error.

Cas d'ús: Consultar estadístiques d'un estudiant

El professor podrà mesurar els coneixements adquirits pels seus estudiants i valorar posteriorment els resultats obtinguts. Els informes generats, donen detall dels testos, la nota obtinguda, les respostes que s'hagin donat, els índexs de resolució i finalització, les notes mitjanes o errors comesos. D'aquesta manera el professor podrà avaluar i treure conclusions que li serviran per a fer reajustaments o bé avaluar el nivell cognitiu dels seus estudiants.

USUARI	SISTEMA
1.El professor es connecta a la pàgina i entra el seu login i password.	2. El sistema registra la informació.
3.El professor selecciona l'assignatura de la qual visualitzar les estadístiques.	4. El sistema registra l'assignatura.
5.El professor selecciona l'opció de visualitzar estadístiques d'un estudiant.	6. El sistema genera una llista dels estudiants de l'assignatura.
7.El professor escull un dels estudiants de l'assignatura.	8. El sistema registra la situació i mostra un informe amb les estadístiques de l'estudiant.
9. El professor visualitza l'informe.	

Casos d'error:

- 1. El login i el password no són correctes. S'ha denegat l'accés a la pàgina i s'informa de l'error.

Cas d'ús: Donar de baixa un anunci

Aquest cas d'ús detalla els passos que ha de seguir el professor per a donar de baixa un anunci del taulell d'anuncis de l'assignatura. El professor accedirà al camp de treball de l'assignatura i indicarà que vol donar de baixa un anunci. Tot seguit elegirà l'anunci que vol donar de baixa i s'eliminarà tal anunci.

USUARI	SISTEMA
1.L'administrador es connecta a la pàgina i entra el seu login i password.	2. El sistema registra la informació.
3.El professor accedeix al camp de treball de l'assignatura on es troba l'anunci que vol donar de baixa.	4. El sistema registra la situació.
5.El professor indica que vol donar de baixa un dels anuncis del taulell d'anuncis.	6. El sistema registra la informació i genera una llista amb els anuncis del taulell.
7.El professor selecciona l'anunci que desitja esborrar.	8. El sistema registra la informació i esborra l'anunci seleccionat del taulell d'anuncis .
9. El professor veu que tot ha anat correctament.	

Casos d'error:

- 1. El login i el password no són correctes. S'ha denegat l'accés a la pàgina i s'informa de l'error.

D.5 Casos d'ús específics dels alumnes

Cas d'ús: Resoldre Test

Aquest és sense cap mena de dubte, la raó principal (no la única), per la qual es desenvolupa l'aplicació. Aquí resollem el problema de com l'estudiant pot dur a terme l'autoavaluació a través de tests.

Un estudiant resol un test aleatori generat per l'aplicació, tindrà un temps determinat per a resoldre'l i al final podrà visualitzar un informe de com li ha anat el test, les respostes que ha fallat, així com les que ha encertat, la puntuació i un comentari per a la resposta escollida.

Si l'alumne no acaba de realitzar el test amb el temps establert, aquest s'emmagatzemarà amb la puntuació que es porti abans d'acabar-se el temps.

Al resoldre el test es generarà una sèrie d'estadístiques que podran ser aprofitades pel professor.

USUARI	SISTEMA
1. L'estudiant es connecta a la pàgina i entra el seu login i password.	2. El sistema registra la informació.
3. L'estudiant selecciona l'assignatura de la qual desitja resoldre el test.	4. El sistema registra l'assignatura.
5. L'estudiant selecciona l'opció resoldre test.	6. El sistema genera una llista dels temes de l'assignatura.
7. L'estudiant escull un dels temes dels quals disposa l'assignatura.	8. El sistema registra el tema i genera una llista dels objectius del tema.
9. L'estudiant escull un dels objectius del tema escollit.	10. El sistema genera automàticament un test de N preguntes aleatòries (dins de les que estan a l'objectiu), guarda l'hora d'inici i ho lliga a un estudiant.
11. L'usuari resol el test (seleccionant una sèrie de respostes associades a cada pregunta del test).	12. El sistema registra al test l'hora de finalització i les preguntes amb la seva puntuació. També es mostra un informe del test, aquest informe inclou cadascuna de les preguntes amb la resposta escollida, un comentari i la correcció.
13. L'usuari visualitza l'informe.	

Casos d' error:

- 1. El login i el password no són correctes. S'ha denegat l'accés a la pàgina i s'informa de l'error.
- 11. L'usuari tanca el test sense finalitzar, el sistema ha de registrar aquesta situació, el test quedaria suspès.

- 11. A l'usuari se li acaba el temps per a poder resoldre el test, el sistema registra el resultat, el temps i les respostes encertades.

Apèndix E

Manual de desplegament

El procés de desplegament de l'aplicació InnovaCampus ens permetrà posar en funcionament la nova aplicació. Per tal d'aconseguir aquest propòsit, s'han de complir una sèrie de requisits i una sèrie de passos que a continuació detallarem.

E.1 Requisits mínims

- Apache Tomcat versió 6 o més.
 - (Servidor web on desplegarem l'aplicació) <http://tomcat.apache.org/download-60.cg>
- PostgreSQL última versió.
 - (SGBD que utilitza l'aplicació) <http://www.postgresql.org/download/>
- Java Development Kit versió 6 o més
 - (Compilador de la part servidor de l'aplicació) <http://java.sun.com/javase/downloads/index.jsp>
- SDK Google Web Toolkit versió 2 o més.
 - (Compilador de la part client de l'aplicació) <http://code.google.com/intl/es-ES/webtoolkit/download.html>
- Client Subversion
 - (Aplicació que ens permetrà descarregar l'aplicació InnovaCampus) <http://subversion.apache.org/packages.html>
- Codi font aplicació InnovaCampus
 - (Aplicació InnovaCampus no compilada)

Un cop tinguem tots els requisits descarregats i instal·lats, ja podem començar a realitzar els passos del desplegament de l'aplicació InnovaCampus, per tal de poder posar-la en funcionament.

Nota: Els processos d'instal·lació que es descriuen a continuació corresponen a instal·lacions estàndards en sistemes operatius Linux, tot i així l'aplicació és compatible amb altres sistemes operatius.

E.2 Instal·lació del servidor Apache Tomcat

Tomcat és un servidor web amb suport de servlets i JSPs. Inclou el compilador Jasper, que compila JSPs convertint-los en servlets. El motor de servlets de Tomcat sovint es presenta en combinació amb el servidor web Apache o altres servidors web. Tomcat pot funcionar com servidor web per si mateix.

Obtenir una còpia del packet d'instal·lació

Per aconseguir una còpia, ens dirigim a l'adreça URL que anteriorment hem citat i en descarreguem un dels paquets, el que més s'adapti a les nostres necessitats i habilitats. En aquest cas nosaltres triem el paquet amb extensió `.tar.gz`.

Procés d'instal·lació

Per tal de descomprimir el packet d'instal·lació de Tomcat que hem obtingut d'internet realitzarem els següent pasos:

1. Crearem una nova carpeta “Servidor” on instal·larem el servidor apache tomcat.
2. Copiarem el paquet descarregat a la carpeta “Servidor”.
3. Descomprimirem el contingut del paquet mitjançant la comanda `tar xzf nom_packet.tar.gz`
4. Comprovarem que el contingut del paquet ha estat descomprimit correctament
5. Borrarem el paquet mitjançant la comanda `rm nom_packet.tar.gz`
6. Executarem al terminal la comanda `sh startup.sh`
7. Obrirem el navegador i executarem la següent URL `http://localhost:8080`
8. Si tot és correcte hem d'obtenir la pàgina d'inici del servidor Tomcat.

E.3 Instal·lació de PostgreSQL

PostgreSQL és un programari lliure que implementa un sistema de gestió de bases de dades objecte-relacional, distribuït sota llicència BSD. Ofereix una alternativa a altres sistemes de gestió de base de dades com ara MySQL, Oracle, DB2 i Microsoft SQL Server entre altres.

Obtenir una còpia del paquet d'instal·lació i instal·lar-lo

Per aconseguir una còpia, ens dirigim a l'adreça URL que anteriorment hem citat i en descarreguem un dels paquets, el que més s'adapti a les nostres necessitats i habilitats.

Procés d'instal·lació

1. Per tal de no tenir cap problema en el procés d'instal·lació, obtindrem permisos de ROOT mitjançant la següent comanda `su -l`

2. Afegim el grup postgres i dins li afegim un nou usuari postgres mitjançant les següents comandes:
groupadd postgres useradd -g postgres postgres
3. Descomprim el paquet mitjançant la següent comanda tar -xvf postgres.x.x.x.tar.gz
4. Accedim a la carpeta on s'ha descomprimit tot el paquet i executem la següent comanda per tal de preparar les fonts a compilar ./configure
5. Compilem les fonts make make install. Si tot ha estat correcte iniciem postgres mitjançant la comanda psql

E.4 Instal·lar JAVA

Per tal de que el nostre ordenador pugui entendre i executar l'aplicació InnovaCampus, necessitem descarregar i instal·lar el kit de desenvolupament de software java, el qual disposa d'un conjunt d'eines que permeten al programador crear aplicacions. Tenint en compte que l'aplicació InnovaCampus ens la descarregarem en codi font, aquesta haurà de ser compilada per tal que el nostre computador la pugui entendre i executar. Aquest procés de compilació el realitzarà una de les eines (el compilador java) que hi ha dins del kit de desenvolupament de software java.

Obtenir una còpia del paquet de desenvolupament java i instal·lar-lo

Per aconseguir una còpia, ens dirigim a l'adreça URL que anteriorment hem citat i en descarreguem un dels paquets, el que més s'adapti a les nostres necessitats i habilitats. La instal·lació la podeu fer seguint els passos i les indicacions que trobareu en el paquet descarregat. Nota: És molt important tenir constància del directori on instal·leu el JDK ja que en passos posteriors requerirem l'adreça d'on es troba instal·lat.

E.5 Instal·lar SDK de Google Web Toolkit

GWT o Google Web Toolkit és un framework creat per Google que permet ocultar la complexitat de diversos aspectes de la tecnologia AJAX. És compatible amb diversos navegadors, la qual cosa és notòria ja que cada navegador sol necessitar codi específic per a assolir un front-end correcte en una aplicació web. El concepte de Google Web Toolkit és bastant senzill, bàsicament el que s'ha de fer és crear el codi en Java usant qualsevol entorn de desenvolupament (IDE) de Java i el compilador ho traduirà a HTML i Javascript.

Obtenir una còpia del paquet de desenvolupament java i instal·lar-lo

Per aconseguir una còpia, ens dirigim a l'adreça URL que anteriorment hem citat i en descarreguem un dels paquets, el que més s'adapti a les nostres necessitats i habilitats. Nota: És molt important tenir constància del directori on instal·leu el SDK GWT ja que en passos posterior requerirem l'adreça d'on es troba instal·lat.

E.6 Instal·lar client subversion

Subversion és un sistema de control de versions dissenyat específicament per reemplaçar el popular CVS, el qual té diverses deficiències. És programari lliure sota una llicència de tipus Apache/Llicència BSD i se'l coneix també com a svn perquè aquest és el seu nom a la línia de comandes.

Obtenir una còpia del packet client subversion

Per aconseguir una còpia, ens dirigim a l'adreça URL que anteriorment hem citat i en descarreguem un dels paquets, el que més s'adapti a les nostres necessitats i habilitats.

E.7 Descarregar i Compilar el codi font de l'aplicació

Per tal d'obtenir el codi font de l'aplicació InnovaCampus s'ha d'executar la següent comanda des del terminal.

```
SVN CHECKOUT http://sedna.udl.cat/innovaGWT/InnovaCampus/
```

E.8 Importació de la BD's de l'aplicació

Abans d'importar crearem una nova base de dades a postgres per tal de poder copiar-hi el contingut de la base de dades de l'aplicació.

```
Create database InnovaCampusDB
```

Per importar la base de dades de l'aplicació a la nova bases de dades que acabem de crear, executarem la següent comanda mitjançant el fitxer .sql que trobarem dins del paquet descarregat a la carpeta db.

```
psql -U postgres -d postgres -f InnovaCampusDB.sql
```

E.9 Configuració de l'aplicació

Quan tinguem el contingut totalment descarregat i la base de dades totalment configurada, modificar una sèrie de línies de diferents fitxers del paquet de l'aplicació descarregat anteriorment.

- Línies 23 – build.xml: `<property name="catalina.home" value="ruta_apache_tomcat"/>` Dins de l'atribut value modificarem el contingut existent, per la ruta real on es troba el nostre servidor apache tomcat.
- Línies 35 – build.xml: `<property name="gwt.sdk" location="ruta_gwt_compilador" />` Dins de l'atribut location modificarem el contingut existent, per la ruta real del directori on es troba el nostre compilador GWT.
- Línies 31 i 32 – applicationContext.xml: `<property name="username" value="postgres" />`
`<property name="password" value="xxxxxx" />`

Dins de l'atribut value modificarem el contingut existent, pel nom d'usuari i el password reals de la nostra base de dades.

- Línies 27 i 28 – testApplicationContext.xml: `<property name="username" value="postgres" />`
`<property name="password" value="xxxxxx" />`

Dins de l'atribut value modificarem el contingut existent, pel nom d'usuari i el password reals de la nostra base de dades.

- Línies 27 i 28 – jdbc.properties: `database.username=postgres database.password=xxxxxx`

Modificarem el contingut existent, pel nom d'usuari i el password reals de la nostra base de dades.

Un cop tinguem modificades correctament aquestes línies procedirem a executar la compilació i posada en funcionament de l'aplicació mitjançant l'execució de la comanda ant, la qual ens compilarà el contingut del paquet i ens exportarà el contingut compilat al servidor per tal de que tot funcioni correctament.

Apèndix F

La llicència GPL

La llicència que utilitzarem per a la nostra aplicació serà la GPL. Aquest annex inclou, doncs, una traducció al català de la llicència GPL, el text original de la qual està en anglès. No ha estat publicada per la Free Software Foundation (fundació per al programari lliure), i no defineix legalment els termes de distribució del programari que utilitza la GPL de GNU (només el text original en anglès ho fa).

LLICÈNCIA PÚBLICA GENERAL DE GNU

Versió 2, Juny de 1991 Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

Es permet que tothom faci còpies literals d'aquest document de llicència i les distribueixi, però no és permès modificar-lo.

F.1 Preàmbul

Les llicències de la major part de programari estan pensades per prendre-us la llibertat de compartir-lo i modificar-lo. Contràriament, la Llicència pública general de GNU pretén garantir-vos la llibertat de compartir i modificar el programari lliure – assegurar-se que el programari sigui lliure per a tots els seus usuaris i usuàries. Aquesta Llicència pública general de GNU afecta la majoria del programari de la Free Software Foundation (fundació pel programari lliure) i a qualsevol altre programa els autors o autores del qual decideixin usar-la (algun altre programari de la Free Software Foundation, en canvi, està protegit per la Llicència pública general per a biblioteques de GNU [NdeT: aquesta llicència, la LGPL, és la que actualment la FSF anomena Llicència pública menys general]). També podeu aplicar-la als vostres programes. Quan parlem de programari lliure (free software) ens referim a la llibertat, no al preu [NdeT: en català no hi ha l'ambigüitat que hi ha en anglès]. Les nostres llicències públiques generals estan pensades per assegurar que tingueu la llibertat de distribuir còpies del programari lliure (i cobrar per aquest servei si així ho voleu), que rebeu el codi font o que el pugueu rebre si el voleu, que pugueu modificar el programari o fer-ne servir parts en programes lliures nous; i que sapigueu que podeu fer aquestes coses. Per protegir els vostres drets, hem d'imposar restriccions que prohibeixin a tothom denegar-vos aquests drets o demanar-vos que hi renunciueu. Aquestes restriccions suposen algunes responsabilitats per a vós si distribuïu còpies del programari o si el modifiqueu. Per exemple, si distribuïu còpies d'un aquests programes, tant si és de franc com per un preu, heu de donar als destinataris els mateixos drets que teniu vós. Us heu d'assegurar que també ells rebin o puguin obtenir el codi font. I els heu de fer saber aquests termes per tal que coneguin els seus drets. Protegim els vostres drets en dos passos: (1) Ens reservem el copyright del programari, i (2) us oferim aquesta llicència que us dóna permís legal per copiar, distribuir i/o modificar el programari. A més a més, per protegir-nos

i protegir l'autor o autora, volem estar segurs que tothom entén que no hi ha cap garantia per a aquest programari lliure. Si algú modifica el programari i el passa, volem que els destinataris sàpiguen que el que tenen no és l'original, per tal que qualsevol problema que haguessin pogut introduir terceres persones no repercuti en la reputació de l'autor o autora original. Finalment, qualsevol programa lliure està constantment amenaçat per les patents de programari. Volem evitar el perill de que els redistribuïdors d'un programa lliure acabin obtenint llicències de patents, i de resultes el programa esdevingui propietat exclusiva d'algú. Per evitar-ho, he deixat clar que de qualsevol patent se n'han d'emetre llicències per a tothom o no emetre'n cap. Les condicions exactes per a la còpia, distribució i modificació són les següents.

F.2 Termes i condicions per la copia, modificació i distribució

0. Aquesta llicència afecta a qualsevol programa o altra obra que contingui un avís del posseïdor del copyright que digui que es pot distribuir sota els termes d'aquesta llicència pública general. D'ara endavant, el "programa" es refereix a aquest programa o obra, i una "obra basada en el programa" voldrà dir el programa o qualsevol obra derivada segons la llei de copyright: és a dir una obra que contingui el programa o alguna part d'ell, ja sigui literalment o amb modificacions o bé traduït a altres llengües. (Per això mateix, les traduccions s'inclouen sense cap limitació en el terme "modificació"). Ens referim a Cada beneficiari de la llicència com a "vós".

Les activitats que no siguin còpia, distribució o modificació no estan contemplades en aquesta llicència, queden fora del seu àmbit. No es restringeix l'acte d'executar el programa, i la sortida del programa queda contemplada només si el seu contingut constitueix una obra basada en el programa (independentment de que s'hagi creat executant el programa). Que aquest sigui el cas o no depèn de què faci el programa.

1. Podeu copiar i distribuir còpies literals del codi font del programa tal i com el rebeu, en qualsevol mitjà, sempre i quan publiqueu en cada còpia, de manera adient i ben visible, una nota de copyright i una renúncia de garantia; manteniu intactes tots els avisos que fan referència a aquesta llicència i a l'absència de garanties de cap mena; i lliureu a qualsevol altre destinatari del programa una còpia d'aquesta llicència juntament amb el programa.

Podeu cobrar un preu per l'acte físic de trametre una còpia i podeu, si així ho voleu, oferir alguna garantia a canvi d'un preu.

2. Podeu modificar la còpia o còpies del programa o qualsevol tros, tot fent una obra basada en el programa, i podeu copiar i distribuir aquestes modificacions o obres sota els termes de la Secció 1 anterior, sempre i quan també compliu les següents condicions:

- a) Heu de fer que els fitxers modificats portin indicacions ben visibles que diguin que heu modificat els fitxers i la data de la modificació.
- b) Heu d'atorgar gratuïtament a totes les terceres parts els termes d'aquesta mateixa llicència sobre la totalitat de qualsevol obra que distribuïu o publiqueu, que completament o en part contingui o sigui un derivat del programa o qualsevol part d'aquest.
- c) Si el programa modificat normalment llegeix instruccions interactivament quan s'executa, heu de fer que quan s'arrenqui per a aquest ús interactiu de la manera més habitual, imprimeixi o mostri un missatge que inclogui una nota de copyright adient i un avís de que no hi ha garantia (o, si de cas, que digui que la garantia l'oferiu vós mateix) i que els usuaris poden redistribuir el programa sota aquestes condicions, i que indiqui a l'usuari o usuària com veure una còpia d'aquesta llicència. (Excepció: si el mateix programa és interactiu però normalment no escriu un missatge d'aquesta mena, la vostra obra basada en el programa tampoc cal que l'escriui).

Aquests requeriments afecten a l'obra modificada com un tot. Si hi ha parts identificables que no estan derivades del programa, i es poden considerar raonablement com a obres independents i separades en si mateixes, aleshores aquesta llicència i els seus termes no s'apliquen a aquelles parts quan les distribuïu com a obres separades. Però quan distribuïu aquestes mateixes parts integrades en un tot que sigui una obra basada en el programa, la distribució del tot s'ha de fer d'acord amb els termes d'aquesta llicència, i els permisos atorgats a altres beneficiaris abasten el tot sencer i, per tant, totes i cadascuna de les parts, independentment de qui les hagi escrites..

Així doncs, la intenció d'aquesta secció no és reclamar o disputar-vos cap dret a codi que hagueu escrit del tot vós mateix. La intenció és més aviat exercir el dret a controlar la distribució d'obres derivades o col·lectives basades en el programa.

A més a més, la simple agregació amb el programa (o amb una obra basada en el programa) d'altres obres no basades en el programa en un volum d'un mitjà d'emmagatzemament o de distribució no posa aquestes altres obres sota l'àmbit de la llicència.

3. Podeu copiar i distribuir el programa (o una obra basada en el programa, segons la secció 2) en forma executable o de codi objecte d'acord amb els termes de les Seccions 1 i 2 anteriors, sempre i quan també feu una de les coses següents:

a) L'acompanyeu amb el codi font complet corresponent, capaç de ser llegit per un ordinador i en un mitjà utilitzat habitualment per a l'intercanvi de programari; o b) L'acompanyeu amb un oferiment per escrit, amb validesa com a mínim fins al cap de tres anys, de subministrar a tota tercera part, i per un preu no superior al que us costi físicament realitzar la distribució, el codi font complet corresponent, que es distribuirà d'acord amb els termes de les seccions 1 i 2 anteriors en un mitjà utilitzat habitualment per a l'intercanvi de programari; o, c) L'acompanyeu amb la informació que hagueu rebut de l'oferiment de distribuir el codi font corresponent (Aquesta alternativa només és permesa per a la distribució no comercial i només si heu rebut el programa en forma executable o de codi objecte amb aquest oferiment, d'acord amb la subsecció b anterior).

El codi font per a una obra vol dir la forma preferida de l'obra per tal de fer-hi modificacions. Per una obra executable, el codi font complet vol dir tot el codi font per tots els mòduls que conté, més tots els fitxers de definició d'interfícies associats si s'escau, més els scripts que es facin servir per controlar la compilació i la instal·lació de l'executable si s'escau. Tanmateix, fent una excepció especial, el codi font que es distribueixi no cal que inclogui res del que normalment es distribueixi (sia en forma binària o de codi font) amb els components principals (compilador, nucli o similars) del sistema operatiu en què s'executa el programa, tret que el component en qüestió acompanyi l'executable.

Si la distribució de l'executable consisteix en donar accés per copiar-lo d'un lloc determinat, aleshores serveix com a distribució del codi font el fet de donar un accés equivalent per copiar el codi font, encara que les terceres parts no estiguin obligades a copiar el codi font en copiar el codi objecte.

4. No podeu copiar, modificar, reemetre llicències, o distribuir el programa si no és de la forma expressa que atorga aquesta Llicència. Qualsevol altre intent de copiar, modificar, reemetre llicències, o distribuir el programa és il·lícit i finalitzarà automàticament els drets que hagueu obtingut d'aquesta llicència. Tanmateix, les parts que hagin rebut de vós còpies o drets d'acord amb aquesta llicència no veuran les seves llicències finalitzades mentre segueixin observant-les estrictament.

5. No esteu obligat a acceptar aquesta llicència, donat que no l'heu signada. Tanmateix, no hi ha cap altra opció que us doni permís per modificar o distribuir el programa o les seves obres derivades. Aquestes accions queden prohibides per la llei si no accepteu aquesta llicència. Així doncs, en modificar o distribuir el programa o les seves obres derivades, esteu indicant que accepteu aquesta llicència per fer-ho, i tots els seus termes i condicions per copiar, distribuir o modificar el programa o obres basades en ell.

6. Cada cop que distribuïu el programa (o qualsevol obra basada en el programa), el destinatari rep automàticament, de qui va emetre la llicència originàriament, una llicència per copiar, distribuir

o modificar el programa sotmesa a aquests termes i condicions. No podeu imposar cap més restricció a l'exercici dels drets que aquí es confereixen. No sou responsable de fer complir aquesta llicència a terceres parts.

7. Si, a conseqüència d'una decisió judicial, una demanda per infracció d'una patent o per qualsevol altra raó (no exclusivament relacionada amb patents), se us imposen condicions (tant si és per ordre judicial, acord, o el que sigui) que contradiuen les condicions d'aquesta llicència, no quedeu excusat de les condicions d'aquesta llicència. Si no us és possible distribuir de manera que satisfieu alhora les obligacions que us imposa aquesta llicència i qualsevol altra obligació pertinent, aleshores resulta que no podeu distribuir el programa en absolut. Per exemple, si una llicència de patent no permetés redistribuir gratuïtament el programa a aquells que hagin rebut còpies de vós directament o indirecta, aleshores la única manera en què podrieu satisfer tant això com aquesta llicència seria abstenir-vos completament de distribuir el programa.

Si qualsevol fragment d'aquesta secció quedés invalidat o no es pogués fer complir en qualsevol circumstància particular, la intenció és que s'apliqui el balanç de la secció, i que s'apliqui la secció en la seva totalitat en altres circumstàncies.

El propòsit d'aquesta secció no és induir-vos a infringir cap patent ni cap altre requeriment del dret a la propietat, o a discutir-ne la validesa; l'únic propòsit d'aquesta secció és protegir la integritat del sistema de distribució de programari lliure, que s'implementa amb pràctiques de llicència pública. Molta gent ha fet generoses contribucions a l'ampli ventall de programari distribuït per aquest sistema refiant-se de l'aplicació consistent del sistema; li pertoca a l'autor, autora o donant decidir si vol distribuir programari per algun altre sistema, i un beneficiari de la llicència no pot imposar aquesta opció.

Aquesta secció pretén deixar del tot clar el que es considera una conseqüència de la resta de la llicència.

8. Si hi ha països que restringeixen la distribució o l'ús del programari, ja sigui per patents o per interfícies sota copyright, el posseïdor del copyright original que posi el programa sota aquesta llicència pot afegir limitacions geogràfiques explícites que excloguin aquests països, de manera que la distribució només quedi permesa dintre dels països no exclosos, o entre ells. En tal cas, aquesta llicència incorpora la limitació com si estigués escrita en el text de la llicència.

9. La Free Software Foundation (Fundació per al programari lliure) pot publicar versions revisades o noves de la llicència pública general de tant en tant. Aquestes versions noves seran semblants en esperit a la versió present, però poden diferir en detalls per tractar nous neguits o problemes.

Cada versió rep un número de versió distintiu. Si el programa especifica un número de versió d'aquesta llicència que li és aplicable i "qualsevol versió posterior", teniu l'opció de seguir els termes i condicions de la versió especificada o de qualsevol versió publicada posteriorment per la Free Software Foundation. Si el programa no especifica un número de versió d'aquesta llicència, podeu triar qualsevol versió que hagi publicat la Free Software Foundation en qualsevol data.

10. Si voleu incorporar parts del programa en altres programes lliures les condicions de distribució dels quals són diferents, escriviu a l'autor per demanar-li permís. Per al programari que està sota copyright de la Free Software Foundation, escriviu a la Free Software Foundation; de vegades fem excepcions per permetre-ho. Prendrem la nostra decisió guiats pels dos objectius de mantenir la condició de lliure de tots els derivats del nostre programari lliure i de promoure la compartició i la reutilització del programari en general.

F.3 Absència de garanties

11. COM QUE LA L·LICÈNCIA DEL PROGRAMA ÉS GRATUÏTA, NO HI HA GARANTIA PER AL PROGRAMA, EN LA MESURA QUE HO PERMETI LA L·LEI APLICABLE. EXCEPTE EL

QUE ALTRAMENT ES DIGUI PER ESCRIT, ELS POSSEÏDORS DEL COPYRIGHT I/O ALTRES PARTS SUBMINISTREN EL PROGRAMA "TAL QUAL" SENSE CAP MENA DE GARANTIA, NI EXPLÍCITA NI IMPLÍCITA, INCLOSES, ENTRE ALTRES, LES GARANTIES IMPLÍCITES DE COMERCIALITZABILITAT I APTITUD PER A PROPÒSITS DETERMINATS. TOT EL RISC PEL QUE FA A LA QUALITAT I RENDIMENT DEL PROGRAMA ÉS VOSTRE. EN CAS QUE EL PROGRAMA RESULTÉS DEFECTUÓS, VÓS ASSUMIU TOT EL COST D'ASSISTÈNCIA, REPARACIÓ O CORRECCIÓ.

12. EL POSSEÏDOR DEL COPYRIGHT, O QUAalsevol ALTRA PART QUE PUGUI MODIFICAR O REDISTRIBUIR EL PROGRAMA TAL I COM ES PERMET MÉS AMUNT NO US HAURÀ DE RESPONDRE EN CAP CAS, TRET DEL QUE REQUEREIXI LA LLEI APLICABLE O ELS ACORDS PER ESCRIT, PER PERJUDICIS, INCLOSOS ELS INCIDENTALS, DERIVATS, ESPECIALS O GENERALS QUE ES DERIVIN DE L'ÚS O DE LA IMPOSSIBILITAT D'ÚS DEL PROGRAMA (INCLOSES ENTRE D'ALTRES LES PÈRDUES DE DADES, LES DADES QUE EL PROGRAMA HAGI MALMÈS, LES PÈRDUES QUE US HAGI PROVOCAT A VÓS O A TERCERS O LA IMPOSSIBILITAT DE QUE EL PROGRAMA FUNCIONI AMB QUAalsevol ALTRE PROGRAMA), FINS I TOT SI AQUEST POSSEÏDOR O ALTRA PART HA ESTAT ADVERTIDA DE LA POSSIBILITAT D'AQUESTS PERJUDICIS.

F.4 Final dels termes i condicions

Com podeu aplicar aquests termes als vostres programes nous:

Si desenvolupeu un programa nou, i voleu que tingui l'ús més gran possible per part del públic, la millor manera d'aconseguir-ho és fer-lo programari lliure que tothom podrà redistribuir i modificar d'acord amb aquests termes.

Per fer-ho, afegiu els avisos següents al programa. El més segur és posar-los al començament de cada fitxer font per transmetre de la manera més efectiva l'exclusió de garanties; i cada fitxer hauria de portar com a mínim la línia de "copyright" i un apuntador que indiqui on es pot trobar la nota sencera. Una línia amb el nom del programa i una idea de què fa. Copyright (C) any nom de l'autor o autora

Aquest programa és lliure; el podeu redistribuir i/o modificar d'acord amb els termes de la Llicència pública general de GNU tal i com la publica la Free Software Foundation; tant se val la versió 2 de la Llicència com (si ho preferiu) qualsevol versió posterior.

Aquest programa es distribueix amb l'esperança que serà útil, però SENSE CAP GARANTIA; ni tant sols amb la garantia de COMERCIALITZABILITAT O APTITUD PER A PROPÒSITS DETERMINATS. Vegeu la Llicència general pública de GNU per a més detalls.

Hauríeu d'haver rebut una còpia de la llicència pública general de GNU amb aquest programa; si no, escriviu a la Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111- 1307, USA.

Poseu-hi la informació que calgui per contactar amb vós per correu electrònic i de paper.

Si el programa és interactiu, feu-lo treure una breu nota com aquesta quan arrenca en mode interactiu:

Gnomovisió versió 69, Copyright (C) any nom de l'autor o autora El Gnomovisió va SENSE CAP MENA DE GARANTIA; premeu 'mostra g' per saber-ne els detalls. Això és programari lliure, i se us convida a redistribuir-lo d'acord amb certes condicions; piqueu 'mostra c' per saber-ne els detalls.

Les instruccions hipotètiques 'mostra g' i 'mostra c' haurien de mostrar les parts escaients de la llicència pública general. Naturalment, les instruccions poden tenir altres noms que no siguin 'mostra

g' i 'mostra c'; fins i tot podrien ser pics amb el ratolí o opcions de menú, o el que li vagi bé al vostre programa.

També hauríeu d'aconseguir que l'empresari per qui trebal·leu (si trebal·leu de programador) o la vostra escola, si és el cas, signin una "renúncia de copyright" pel programa, si s'escau. Aquí teniu un exemple, canvieu-hi els noms:

Ioiodina, S.A. , per la present renuncia a tot interès en el copyright del programa 'Gnomovisió' (que fa l'aleta als compiladors) escrit pel Jordi Pica Codi

signat per Mag Nat, 28 de desembre de 1989 Mag Nat, Vici-President

Aquesta L·licència pública general no us permet incorporar el vostre programa en programes de propietat. Si el vostre programa és una biblioteca de subrutines, potser trobeu més útil permetre enllaçar (link) aplicacions de propietat amb la biblioteca. Si és això el que voleu, feu servir la L·licència pública general per a biblioteques de GNU per comptes d'aquesta l·licència [NdeT: la FSF ara li diu l·licència pública menys general per comptes de l·licència pública general per a biblioteques].

Acrònims

- AJAX Asynchronous JavaScript And XML
- ASP Active Server Pages
- BSD Berkeley Software Distribution
- CGI Common Gateway Interface
- DTD Document Type Definition
- FSF Free Software Foundation
- FTP File Transfer Protocol
- GNU GNU is Not Unix
- GPL General Public License
- GWT Google Web Toolkit
- HTML Hypertext Markup Language
- HTTP Hypertext Transfer Protocol
- JDBC Java DataBase Connectivity
- JMX Java Management Extensions
- JSP Java Server Pages
- LGPL Lesser General Public License
- MD5 Message-Digest Algorithm 5
- PHP PHP: Hypertext Preprocessor
- RIA Rich Internet Applications
- SGML Standard Generalized Markup Language
- SHA1 Secure Hash Algorithm – version 1.0
- SQL Structured Query Language
- SSJS Server Side JavaScript
- SSL Secure Socket Layer
- TCP/IP Transport Control Protocol/Internet Protocol

- URL Universal Resource Locator
- WWW World Wide Web
- XHTML eXtensible Markup Language
- XML eXtensible Markup Language

Bibliografia

- [1] Marcel·lí Alet Alís. InnovaCampus: Una aplicació Web d'ajut a l'autoavaluació en una titulació universitària. 2006
- [2] Xavier Aiguabella Guilera. Disseny i implementació del mòdul de generació de tests d'una aplicació web d'autoavaluació desenvolupada en comunitat. 2007
- [3] Ramon Xuriguera Albareda. Disseny i implementació d'un mòdul d'estadístiques per a una aplicació d'autoavaluació desenvolupada en comunitat. 2007
- [4] Xavier Aiguabella Guilera, Natàlia Sans Solsona. InnovaCampus 2.0: Redisseny d'una aplicació web d'ajut a l'autoavaluació formativa. 2009
- [5] Hanna, P.: JSP Manual de Referencia. McGraw-Hill. 2002
- [6] Patzer A.: JSP. Ejemplos Prácticos. Ed. Anaya Multimedia. 2003
- [7] Galbraith B., Falknet J., Irani R.: Desarrollo Web con JSP. Ed. Anaya Multimedia. 2002
- [8] Kennedy B., Musciano C.: HTML: The definitive guide. O'REILLY. 1998
- [9] Gutiérrez, A., Martínez, R.: XML a través de ejemplos. RA-MA. 2001
- [10] Ribó, J.M.: Apunts sobre JSP de l'Assignatura Tecnologies actuals de la Programació. UDL. 2004
- [11] Robert Hanson i Adam Tacy. GWT in Action. Manning 2008
- [12] Java Sun.: The Java Tutorial. 2005 <http://java.sun.com/docs/books/tutorial/>
- [13] Java Sun.: JavaTM 2 Platform Standard Edition 5.0 API Specification. 2004 <http://java.sun.com/j2se/1.5.0/docs/api>
- [14] The Apache Software Foundation: Commons DBCP. 2001-2005 <http://jakarta.apache.org/commons/dbcp/apidocs>
- [15] Flanagan. D.: JavaScript: The Definitive Guide, 4th Edition. O'Reilly. 2001 (accedit des de <http://proquest.safaribooksonline.com>)
- [16] Pulido, E.M. , Domínguez, O. , Jiménez, C.: Tutorial de HTML. http://gias720.dis.ulpgc.es/Gias/Cursos/Tutorial_html/indice.htm
- [17] Varis Autors: Wikipedia, la enciclopedia libre. <http://es.wikipedia.org/wiki/Portada>
- [18] The PostgreSQL Development Team: PostgreSQL Manual. 2006 <http://www.postgresql.org/docs/manuals>

- [19] González, O.: Filtros y Servlets en Java. 2003 http://www.samelan.com/oscardgonzalez/doc/java_filters.pdf
- [20] Aparicio, A. L.: Primeros pasos con Tomcat. 2003 <http://bulma.net/body.phtml?nIdNoticia=1888>
- [21] Sun Microsystems: Core J2EE Patterns. 2001-2002 <http://java.sun.com/blueprints/corej2eepatterns/Patterns>
- [22] Asociación española de Internet: Patrones de Diseño en aplicaciones Web con Java J2EE. 2004 http://java.ciberaula.com/articulo/disenio_patrones_j2ee
- [23] Ryan Dewsbury: Google Web Toolkit Applications, Prentice Hall 2008
- [24] MD5. <http://en.wikipedia.org/wiki/MD5>
- [25] SHA hash functions. [http://en.wikipedia.org/wiki/SHA hash functions](http://en.wikipedia.org/wiki/SHA_hash_functions)
- [26] Spring. <http://www.springsource.org/>
- [27] Spring Documentation. <http://static.springsource.org/spring/docs/3.0.x/spring-framework-reference/html>
- [28] Spring per aplicaciones .NET. <http://www.springframework.net/>
- [29] Spring Security. <http://static.springsource.org/spring-security/site/>
- [30] Subversion. <http://subversion.tigris.org/>
- [31] Apache Tiles. <http://tiles.apache.org/>
- [32] The Java Persistence API. <http://java.sun.com/developer/technicalArticles/J2EE/jpa>
- [33] Internationalization and localization. [http://en.wikipedia.org/wiki/Internationalization and localization](http://en.wikipedia.org/wiki/Internationalization_and_localization)
- [34] Antonio J. Martín Sierra: AJAX en J2EE. RA-MA. 2007
- [35] Using the Spring GWTController. <http://sites.google.com/site/ggeorgovassilis/usingthespringgwtcontroller>
- [36] Integrate Spring 2.x with GWT. https://docs.google.com/Doc?docid=dw2zgx2_25492p5qxfq&hl=en
- [37] GWT and SpringSecurity. <http://seewah.blogspot.com/2009/06/gwt-and-spring-security-sample-demo.html>